Malware Communication and Containment in Critical Infrastructure Networks

Theoretical Models, Simulations and Defensive Solutions

Dissertation

for obtaining the academic degree

Dr. techn.

as part of the study

Electrical Engineering and Information Technology

carried out by

Peter Eder-Neuhauser, MSc
student number: 1329364

Institute of Telecommunications
at TU Wien

Supervisor:
Univ.-Prof. Dipl.-Ing. Dr.-Ing. Tanja Zseby

Reviewer:
Univ.-Prof. Dipl.-Ing. Dr.-Ing. Wolfgang Gawlik
Prof. Dipl.-Ing. Dr.-Ing. Georg Carle

Statement of Academic Integrity

I hereby declare and confirm with my signature that the doctoral dissertation is exclusively the result of my own autonomous work, based on my research and literature published, which is seen in the notes and bibliography used. I also declare that no part submitted has been made in an inappropriate way, whether by plagiarizing or infringing on any third person's copyright. Finally, I declare that no part submitted has been plagiarized for any other paper in another higher education institution, research institution or educational institution.

Vienna, February 2018

Author's signature

Acknowledgements

I would like to express my special appreciation to my advisor Univ.-Prof. Dipl.-Ing. Dr.-Ing. Tanja Zseby, who has been a tremendous mentor for me. I am grateful for her continuous efforts in encouraging my research and for allowing me to grow as a scientist. Her advice on both research and career have been priceless. I would also like to thank my committee members, Univ.-Prof. Dipl.-Ing. Dr.-Ing. Wolfgang Gawlik and Prof. Dipl.-Ing. Dr.-Ing. Georg Carle for their valuable input on my work. The remarks made upon my work have allowed me to increase the quality of my dissertation.

A special thanks goes out to my colleagues for the enduring support in a great number of situations when I thought all is lost. Primarily, I would like to express my appreciation toward Dipl.-Ing. Gernot Vormayr and Philip Pausakerl. They have supported my work with malware simulation and also supported my network topology simulation models. Additional thanks goes to Dipl.-Ing. Mag.rer.soc.oec. Dr. techn. Joachim Fabini for his continous support in honing the scientist within me, and to my industrial advisor Dipl.-Ing. Wolfgang Wais for his valuable inputs and data sets.

Even more special thanks go to my family. Words cannot express how grateful I am for all the sacrifices that you have made on my behalf. I would also like to thank all of my friends who supported me. I would like to express my deepest appreciation to my beloved partner Dr. techn. Irene Schöner, MSc who spent sleepless nights on my side and was supportive in the moments when there was no one to answer my queries. This dissertation would not have been possible without you.

Finally, I would like to acknowledge the projects URBEM and SIMULTAN, funded by Wiener Stadtwerke and the Austrian Research Funding Association (FFG), that made this dissertation possible. I would like to extend a special thanks to all supportive staff within the TU and the Wiener Stadtwerke that worked so hard in administrative roles.

Abstract

Critical infrastructures utilize information technology for control functions, which creates additional entry points in vulnerable hard- and software, providing distribution paths for cyber-attacks. In this dissertation we address the issue of cyber-attacks against critical infrastructures in five parts.

First, we provide an evaluation of four network architectures suitable for critical infrastructures. Their security by design and their applicability toward real world scenarios are also considered. We summarize the benefits and drawbacks with a focus on the implementation of self-organizing structures within decentralized and centralized network topologies, regarding security.

Then, we investigate malware communication in critical infrastructures, proposing a comprehensive generic model for cyber-attack life-cycles and addressing the specific characteristics of the environment. We include the building blocks for many major known malware types as well as different propagation methods, access vectors, scanning techniques, command and control structures, attack methods, triggers, and cleanup mechanisms. Toward this end, we evaluate a variety of malware types as basis for our attack model and introduce three novel superclasses that are particularly suited for attacking critical infrastructures. These synthetic models provide a basis for the detection of malware communication and extrapolates from existing malware technologies in order to predict future developments.

Based on these malware models, we conduct discrete-event simulations in the ns3 environment, which are based on our network topologies that use real infrastructure data from our industrial partner. Our investigations show that aggressive malware, although quickly spreading, leaves footprints for defensive mechanisms to effectively counteract them. However, stealthy malware that is less visible and therefore harder to detect, spreads slower but requires more scrutiny on the defenders' side.

We also develop metrics that evaluate the security by design of each network topology and the malware movement inside critical infrastructure networks. We design those metrics to represent malware spreading and consider the importance of critical nodes inside each topology. This allows us to evaluate how different malware types behave from our simulation results and conclude how to defend against them.

Finally, we introduce a list of defensive measures, categorized by functionality and attack type. We correlate these categories to the attack stages that occur during a cyber-attack and map them to our generic cyber-attack life-cycle model.

"We are seeing a democratization of tactics. We used to be able to tell the attacker by the weaponry, as only military could afford tanks. This no longer applies to cyber-space as everyone, nation states and civilian aggressors alike, are using the same tactics, techniques and tools. This situation makes defense policy difficult, as you cannot react accordingly to an unknown attacker."

Bruce Schneier, RSA Conference 2015, San Francisco, CA, USA

Contents

List of Figures

List of Tables

Acronyms

AMI	Advanced Metering Infrastructures
APTs	Advanced Persistent Threats
ARP	Address Resolution Protocol
ASLR	Address Space Layout Randomization
BGP	Border Gateway Protocol
C2C	Client-to-Client
C2S	Client-to-Server
C&C	Command-and-Control
CDC	Centers for Disease Control and Prevention
COSEM	Companion Specification for Energy Metering
CSA	Cyber-Situational-Awareness
DCS	Distributed Control Systems
DDoS	Distributed Denial-of-Service
DEP	Data Execution Prevention
DMZ	Demilitarized Zones or Perimeter Networks
DNP3	Distributed Network Protocol
DNS	Domain Name System
DoS	Denial-of-Service
DSL	Digital Subscriber Line
E2E	End-to-End
EMET	Enhanced Mitigation Experience Toolkit
ENTSO-E	European Network of Transmission System Operators for Electricity
FDMA	Frequency Division Multiple Access
FTP	File Transfer Protocol
H2H	Human-to-Human
H2M	Human-to-Machine
H2NS	Host-to-Network-Share
HAN	Home Area Network
HL-nodes	High-Level-nodes
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
HV	High Voltage
ICMP	Internet Control Message Protocol
ICS-CERT	Industrial Control Systems Cyber Emergency Response Team
ICT	Information and Communication Technology
IDS	Intrusion Detection Systems
IoT	Internet-of-Things
IP	Internet Protocol
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
IRC	Internet Relay Chat
KDC	Key Distribution Center
LL-nodes	Low-Level-nodes
LV	Low Voltage
M2M	Machine-to-Machine
malware	Malicious Software
MAN	Metropolitan Area Network
MitM	Man-in-the-Middle
ML-nodes	Medium-Level-nodes

MMS	Manufacturing Message Specification
MV	Medium Voltage
NAN	Neighborhood Area Network
NAS	Network-attached Storage
NASPI	North American Synchro-Phasor Initiative
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
OLSR	Open Link State Routing
ONS	Object Naming Service
OTN	Optical Transport Network
P2P	Peer-to-Peer
P2P	Point-to-Point
PKI	Public Key Infrastructure
PLC	Power Line Communication
PMU	Phasor Measurement Units
PON	Passive Optical Network
PtH	Pass-the-Hash
PTP	Precision Time Protocol
RD	Removable Drives
RTT	Round-Trip Time
RTU	Remote Terminal Units
S2C	Server-to-Client
S2S	Server-to-Server
SCADA	Supervisory Control and Data Acquisition
SGAM	Smart Grid Architecture Model
SIMULTAN	Simultaneous Planning Environment for Buildings in Resilient, Highly Energy Efficient and Resource-Efficient Districts
SMB	Server Message Block
SONET	Synchronous Optical Network
SPF	Sender Policy Framework
SSH	Secure Shell
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TSO's	Transmission-Grid System Operators
TTL	Time-to-Live
TU Wien	Vienna University of Technology
UDP	User Datagram Protocol
UHV	Ultra High Voltage
URBEM-DK	Urban Energy- and Mobilitysystem
USB	Universal Serial Bus
VLAN's	Virtual Local Area Networks
VPN	Virtual Private Network
WAMS	Wide Area Monitoring Systems
WAN	Wide Area Network
WDMA	Wavelength Division Multiple Access
WiMax	Worldwide Interoperability for Microwave Access
WPAN	Wireless Personal Area Network
WSTW	Wiener Stadtwerke Holding AG

1 Introduction

This work examines information security issues in communication networks that are used to remotely control hardware in critical infrastructures. Due to the increasingly hostile environment, attacks against Information and Communication Technology (ICT) have to be taken more seriously than ever. Highly networked critical infrastructures, such as smart power grids, deserve our special attention in terms of resilience and defendability against motivated and even highly financed adversaries. Effective deterrents against those attackers include, high defensive capabilities, e.g., when the attack does not have the desired impact, and judicial consequence, e.g., when identifying the attacker leads to government involvement. Considering these basic deterrents, high defensive capabilities seem to be the most pressing one because attacks over communication networks are often not identifiable, thus, legal action may not represent a viable option.

Smart grids depend on ICT for managing power flux and energy balance. Additionally, they host many different types of devices, including but not limited to measurement equipment (e.g., Phasor Measurement Units (PMU) and smart meters), actuators (e.g., breaker-switches and disconnectors), and networking equipment (e.g., gateways, servers, and routers). These critical devices are just as susceptible to Malicious Software (malware) as are classical Internet technologies such as, e.g., laptops, smartphones, and other consumer electronics. However, power grid environments have a focus on long-term stability and plan for hardware life-spans of 10 years or more. As devices age, unknown vulnerabilities in hardware, operating system, software, and protocols may be found. Although ICT allows for new remote control capabilities increasing the management efficiency of utilities, it also increases their attack surface. This issue must be addressed in order to ensure the security of future energy networks [48].

Today, few malware implementations that caused severe physical damage to industrial equipment are known. However, during the last decade the number of such highly evolved malware capable of orchestrated cyber-physical attacks has increased. Their sophistication leads to the assumption that massive resources were invested and that they may be financed by highly capable stakeholders, e.g., organized criminal collectives, or nation states [56,94,123]. The detection and analysis of existing malware is of paramount importance for the implementation of defensive measures.

1.1 Motivation

The motivation of working on this dissertation in such a fast growing field descends from four main-pillars which come together at the intersection where two critical infrastructures meet, namely, the electricity-grid and communication networks. The automation technology required in electricity grids provides a number of benefits, that also come with drawbacks inherited from the Internet landscape. Our motivations include:

- The *critical nature* of electricity grids to modern society, and the *interconnectivity* with other critical infrastructures leads to interdependabilities that must be managed. Appendix A illustrates a poster that shows many examples of society's dependency to critical infrastructures. This poster has an educational background [42]. [60, 64]
- The increasing *trend of automation* in the energy-industry leads to increased *attack surface* of critical infrastructures which can expose them to cyber-attacks.
- The increasingly *complex environment* in terms of interdependencies leads to complex attack-vectors and failure-scenarios which need to be addressed in emergency response plans.
- The increasing number and *complexity of cyber-attacks* posing a serious threat to critical infrastructures.

1.2 Objective

The goal of this dissertation is to investigate malware communication, propagation and attack containment in critical infrastructure networks, specifically tailored to smart grid operations. Aside from malware behavior [48], e.g., propagation mechanisms, attack techniques, and scanning methods, we put forward evaluation metrics on malware movement [46], security by design for network topologies [44], and defensive measures [43] for utility providers. Among those different critical infrastructures [42], we concentrate on power grids and communication networks, because they are the most relevant in terms of interconnectivity and attack propagation. Additionally, the most immediate effects become apparent once electricity supply is impaired. Social implications and their repercussions toward the public are out of scope.

Furthermore, our list of defensive measures, cf. Section 10, represents a valuable asset for educational purposes. Since we cannot claim completeness of this list, our defensive measures are out of scope for the simulation model. Instead we focus on malware simulations that include the scanning and propagation behavior in four different network topologies and how defendable each setup is. Therefore, grid operators should choose the corresponding defensive measures as they are subject to specific technical circumstances, personal preference, and available resources. Since critical infrastructures provide for many of our basic needs as a society, we stress their importance and focus our efforts on their protection.

1.3 Research Questions

This work investigates malware propagation in communication networks of smart grid control systems and identifies relevant measures on corresponding security implications that are used to contain the potential damage of cyber-attacks. The research questions are as follows, cf. Figure 1:

1. **How can a communication topology deliver increased security by design?**

   Communication networks can never be 100% secure. Therefore, we aim to investigate if and how the security by design of certain network topologies can be increased, and what measures must be taken to accomplish that. Our goal is to discover effective ways to contain network propagation from infected host. The output of research question 1 is an in-depth theoretical comparison of four ICT topologies regarding security properties, counter measures, and legacy system-compatibility. This initial analysis prepares the basis for continued simulations, cf. Section 9.3, and for research question 2.

2. **How does malware propagate inside critical infrastructure communication networks?**

   We investigate several malware models, ranging from aggressive to stealthy types, cf. Section 8.1, to find out how fast and efficient different malware spreads in different environments. Furthermore, our initial investigation concerning, e.g., scanning type, propagation method, connection attempts, protocols, or spreading patterns, shows that these metrics have considerable influence on the infection rates. Our goal is to develop effective detection methods for different types of malware. Furthermore, we discuss effective containment measures against cyber-attacks. The output of research question 2 is a theoretical comparison of existing malware types, propagation vectors, attack types, scanning techniques, update mechanisms, defensive measures, and covert techniques for critical infrastructure network environments. Furthermore, we introduce a novel generic model of a malware life-cycle analysis, three generic malware models, and an outlook on smart grid specific attacks. Our three malware types are simulated across the four network topologies developed during the work of research question 1.

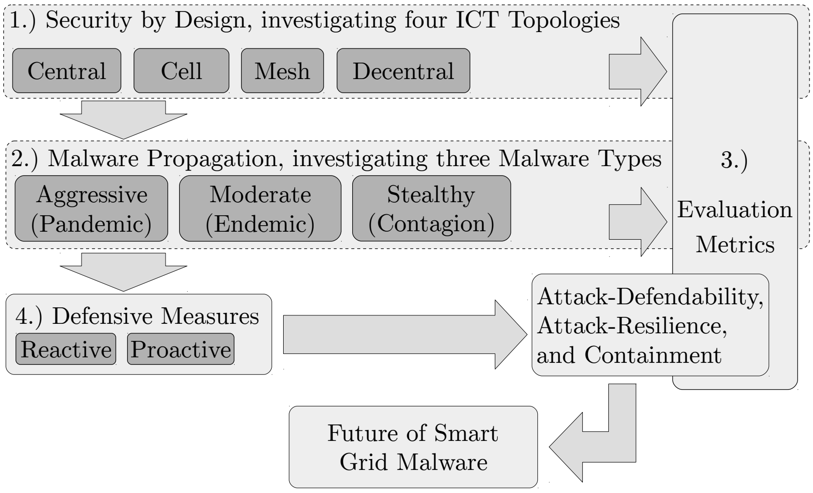

1.) Security by Design, investigating four ICT Topologies
Central
Cell
Mesh
Decentral
2.) Malware Propagation, investigating three Malware Types
Aggressive (Pandemic)
Moderate (Endemic)
Stealthy (Contagion)
3.) Evaluation Metrics
4.) Defensive Measures
Reactive
Proactive
Attack-Defendability, Attack-Resilience, and Containment
Future of Smart Grid Malware

Figure 1: Relations between the research questions

3. **How to assess the properties of different malware types in smart grid communication networks?**

   Based on our initial malware investigation in research question 2, we develop several metrics that describe the behavior of different malware types inside our critical infrastructure network environments, developed in research question 1. We introduce these metrics with the goal of formalizing our malware simulation models with all their properties, e.g., network scanning, payload propagation, infection ratios, infection speed, stealthiness, and topological features. The output of research 3 includes a set of evaluation metrics that represent malware properties relevant for detecting cyber attacks.

4. **How to increase the attack-defendability and attack-resilience of critical communication networks?**

   We introduce the metric *attack-defendability* as a measure for security by design, cf. Section 6.13, that can be calculated with our simulation model. Furthermore, we define the metric *attack-resilience* as the resistance of a network against intentional attacks and expand our list of defensive measures upon additional defense measures. Attack-resilience represents that cyber-attacks do not occur randomly, but are targeted, and thus, represent deliberate failures. Therefore, we cannot rely on redundancies, but instead on proactive and reactive counter measures, cf. Section 10. The output of research question 4 is a collection of proactive and reactive measures to achieve a higher level of security.

1.4 Research Methods

First, we conduct an extensive *investigation on network topologies* suitable for smart grid environments, cf. research question 1, to define our working environment and extract important features that impact malware movement inside those topologies.

Then, we conduct an extensive *investigation on existing malware types* which are suitable for attacks against smart grids, cf. research question 2. We extract the most important features on malware communication, propagation, scanning, and attack vectors that can be used against smart grids. These investigations include theoretical considerations and literature research. Additionally, we extract suitable evaluation metrics for both, the network topologies and the malware types, which are used in our simulation model.

Next, we develop a *simulation* environment for the *analyses* of network topologies and malware capabilities based on ns3. This simulation environment is evaluated with the *metrics*, cf. research question 3, that describe the malware behavior for different network topologies.

Finally, a list of *defensive measures*, which is developed throughout the course of this work, represents our solution for malware spreading in critical infrastructure networks, cf. research question 4, as presented in the results section.

Table 1: Research Questions and Methods

Research question	Methodology	Key finding
1. Network topology	Theoretical investigation	The *cell topology* with its well segmented sub-networks shows the greatest benefit against highly developed (APT) malware, which are most difficult to defend against.
1. Network topology	Simulations	When considering that the defensive measures are implemented to state-of-the-art level, the cell topology provides the best defensive characteristics due to its neuralgic nodes at strategically important points.
2. Malware propagation	Theoretical investigation	Malware shows increasing modularity which in many cases adds *stealthiness* as the most significant benefit at the cost of high propagation speed.
2. Malware propagation	Simulations	Our simulations show simple malware propagates fastest in centralized topologies, but is easiest to defend against. Whereas in mesh networks, providing additional distribution paths, sophisticated malware exhibits benefit by *advanced stealthiness features*. We add new metrics to mathematically substantiate the results.
3. Malware metrics	Theoretical investigation and simulation	Our malware metrics represent the malware speed, stealthiness, and infection ratios that are used to classify our three malware types in our four network topologies. Furthermore, we extend our calculable metrics upon additional *metrics to support future work.*
4. Attack-defendability	Theoretical investigation	An extensive list of *defense measures* is included in the results section.
4. Attack-defendability	Simulations	Our simulations confirm the results of our theoretical investigation. *Increased stealthiness* adds the most benefit for attackers, thus, requires the most defense effort.

1.5 Contributions

In this section we list the major contributions of this work:

- We investigate basic assumptions on the electricity grid hierarchy and corresponding network topologies, using real data from the city of Vienna, outlining differences between the energy- and ICT domain in an extensive analysis. These topology variants range from fully centralized to fully decentralized types and hybrid concepts [44].
- We introduce a generic malware-based attack-life-cycle that represents all aspects from our extensive investigation on existing malware. Furthermore, we use this attack-life-cycle model to extract the most important features in our malware investigation the most important features used in smart grid attacks [48].
- From this attack-life-cycle model we develop three malware superclasses that represent a number of attack vectors, currently common malware features, and new features that are on the rise. They represent our malware model against smart grid attacks [48].
- We introduce a number of metrics for evaluating malware detection, propagation, and scanning behavior, whilst infection duration metrics represent the defendability in our network topology models [46].
- We develop an ns3-based simulation environment that represents the malware behavior [48], modeled by the author and programmed in cooperation with a diploma thesis (work in progress). Additionally, our ns3 environment includes the network topologies, that represent our smart grid networks. The simulations conducted with this model represent our propagation and defendability study.
- We maintain a list of defensive measures that can aid utilities to be implemented against cyber-attacks, cf. [41]. The measures are categorized by the malware superclasses and compared to the simulation results. Additionally, we published an educational poster on the interdependencies of critical infrastructures, cf. Appendix A [42], including many examples. These examples should bring this complex topic closer to the general public.

1.6 Structure

We structure this work as follows.

Network Topology Models:

An analysis of network topologies suitable to the requirements of critical infrastructures represents our basic network environment. We provide a qualitative evaluation of four ICT topologies based on existing smart grid reference architectures that support the existing power transmission and distribution hierarchy. We analyze the benefits and drawbacks of each ICT topology in an urban context, derived from the existing reference architectures with a focus on security by design [44].

Malware Models

Furthermore, we introduce a generic malware-based attack-life-cycle model that formalizes many existing malware types, thus, represents our basic generic malware model, which allows us to build the simulation models [48]. Based on our generic life-cycle model and the investigation of existing malware we create three malware superclasses, cf. Section 8.2, that represent different levels of attack-sophistication against the operators of critical infrastructures.

Metrics

We define new metrics to evaluate the malware stealthiness, propagation vectors, infection ratios, detection, and defendability characteristics of each malware type. Additionally, we investigate the metrics with different malware types in the simulation model and extend and prepare our metrics for future works [46].

Simulation Environment

Based on the smart grid network environment and the malware models we simulate cyber-attacks in the ns3 network simulation environment to analyze the malware communication and propagation [46].

Defense Measures

We conclude our work with a list of recommendations to be implemented by utility companies, cf. Section 10. These defensive measures should provide a reasonable level of security for critical infrastructures and enable utilities to mitigate malware based cyber-attacks [43].

1.7 Terminology

cf. research question 3 In this section we discuss the naming conventions and frequently used words.

- This work generally discusses issues at the boundary of electricity- and communication-technology, hence cyber-physical systems. However, our focus rests on ICT networks, yet we consider many examples with implications on the electricity grid. First, demarcations between the electricity grid and ICT network have to be noted. The term "*grid*" henceforth only refers to energy grid related issues. Consequently, the term "*network*" concerns only communication technologies.
- The term "*architecture*" is used synonymously with "*topology*" and involves all matters of ICT or power-grid design. However, other disciplines of research may understand the term "architecture" from a construction-engineering point-of-view. To prevent any confusion, we clarify that the remainder of this work applies "architecture" only for network design, e.g. star topology or mesh network topology.
- Concerning "hierarchy" we follow the Smart Grid Architecture Model (SGAM) principle [21] and base our wording on their power grid hierarchy. Therefore, ICT nodes concerning high level functions in the high voltage grid are called high level nodes, medium level node have medium level functions, and so on. We elaborate on naming conventions and node types in Section 3.3.
- Different types of malware, cf. [18,114,151,166], are capable of utilizing functional malware such as trojan horses, spyware, adware, spammers, sniffers, crypto-lockers, backdoors, logic bombs, and others in modular extensions. Although they behave vastly different, at some point they utilize a payload that exploits a vulnerability. However, for the remainder of this work we will use the term *"malware"* to refer to all malware-types simultaneously. For instance, worms often extend their functionality by downloading modules, which could be seen as its own malware type. However, it is not useful for our work to categorize them into any more detail, as shown in [13, 18, 123].
- The terms "attack-resilience" and "resilience" differ in meaning because the latter is heavily used in literature to describe equipment failure that occurs without malicious intent. However, "attack-resilience", a term coined in our metrics, includes malicious intent and defines how well a system can be defended against planned attacks. Both terms are used in their respective context throughout this work.
- We often mention *field nodes*, which includes all low-level nodes (e.g., smart meters) and medium-level nodes (e.g., gateways).

1.8 Support

This dissertation was done as part of the doctoral college URBEM. We list all supporting bodies below.

TU Wien - Institute of Telecommunications

The Institute for Telecommunications at Vienna University of Technology (TU Wien) is a competence center for ICT systems at the faculty of Electrical Engineering and Information Technology. The Communication Networks group is working in the field of reactive security solutions for the timely, reliable data transmission with high integrity and reliability. One of the key issues is the monitoring and security by design of cyber-physical systems.

Wiener Stadtwerke Holding AG

The Wiener Stadtwerke Group is among the 25 largest companies in Austria and makes an important contribution to the functioning of the city of Vienna. With their consolidated corporate sectors of power-, gas-, heat-, and water-grids, energy providers, communication networks, public transport, parking-space management, and cemeteries, the Wiener Stadtwerke Group employs approximately 16,100 persons. This industrial partner invests substantial funds in research for an ever increasing standard of living.

URBEM

The Wiener Stadtwerke Holding AG (WSTW), a public utilities company and the TU Wien have together instantiated a Doctorate College entitled Urban Energy- and Mobilitysystem (URBEM-DK). The aim is to research and develop an interactive environment for analyzing scenarios for a "sustainable, supply oriented, secure, affordable and livable city" by the example of Vienna. In a holistic and interdisciplinary approach (keyword "Smart City").

SIMULTAN

Also instantiated by WSTW and the TU Wien, the SIMULTAN project is researching sustainable buildings in the context of urban districts. The acronym stands for Simultaneous Planning Environment for Buildings in Resilient, Highly Energy Efficient and Resource-Efficient Districts (SIMULTAN). This project only considers the building level, which we represent in the lowest hierarchy level of the simulation environment.

2 State of the art

Notice of adoption from previous publications:
Parts of this chapter have been previously published in:

- *[44] P. Eder-Neuhauser, T. Zseby, and J. Fabini. Resilience and Security: A Qualitative Survey of Urban Smart Grid Architectures.* IEEE Access*, 4:839–848, doi: 10.1109/ACCESS.2016.2531279, February 2016*
- *[48] P. Eder-Neuhauser, T. Zseby, J. Fabini, and G. Vormayr. Cyber Attack Models for Smart Grid Environments.* Sustainable Energy, Grids and Networks*, (12C (2017)):10–29, doi: 10.1016/j.segan.2017.08.002, August 2017*

The theoretical base for our ICT topology study was introduced in [44]. It outlines differences between the energy domain and the ICT domain. Several parts of the general discussion, figures and state-of-the-art smart grid topologies were presented. The main contribution of the author includes an in depth analysis of network architectures suitable for smart grid operations, while the co-authors added valuable input on additional details on network technology in discussions. These models were also presented in [37, 39].

[48] discusses several malware-types capable of attacking smart grid environments. Several parts of state-of-the-art malware was discussed in this chapter. The main contribution of the author was an extensive analysis of several malware types and a discussion of the threat level of different attackers. The co-authors contributed valuable additional input in discussions. A summary of the malware capabilities was also presented in [38, 40, 45].

This section discusses the current state of the art in energy grids (particularly the power grid), power transmission, power distribution, advanced metering infrastructures, wide area monitoring systems, threats to smart grids, and malware capabilities. Furthermore, we discuss the differences between smart grid and Internet technologies, including their impact on future power grids.

2.1 Energy Grids

Energy grids include electricity-, heat- and gas-grids, which are discussed in the following sections. However, we concentrate our efforts on electricity grids and their communication infrastructures because we find that they have the most immediate effect in the event of an attack or large scale failure compared to heat- and gas-grids, which have greater inertia to changes.

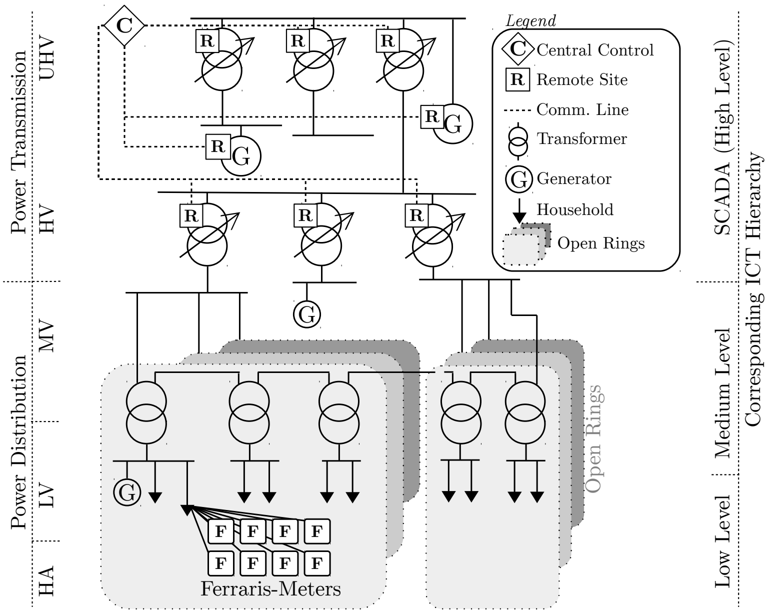


Figure 2: The power grid today, as presented in [44]

2.1.1 Electricity Transport, Micro-, and Smart Grids

The European *power transmission grid* is a synchronous power grid that spans 34 countries and is organized by the European Network of Transmission System Operators for Electricity (ENTSO-E), a network of numerous Transmission-Grid System Operators (TSO's) [52, 188]. It was built using a centralized approach with a small number of large generation units at the highest level, while consumption is subsumed within the numerous distribution grids allocated subjacent to high-voltage substations. The transmission grid's purpose is primarily transporting and balancing energy across vast distances in different countries and regions.

Distribution grids connect to the transmission grid via substations, cf. Figure 2. The main purpose of power distribution is to deliver energy to the customers. Kerber and Witzmann [102] argue that most urban distribution grids are organized in open rings, allowing alternative reconnection routes to circumvent faulted parts. In recent years numerous decentralized renewable energy generators have been installed at the *power distribution* level, following the policies on clean energy. These small generators have the potential to push fossil power generation out of the energy market at times, leading to a decrease in rotating masses (operating bulk generators) in the transmission grid. However, unpredictable decentralized generation could result in voltage- or even frequency-fluctuations. Conventional power generation thus serves as an emergency reserve in case renewable power generation falls short. In view of the rising number of decentralized installations, grid operators increasingly require that renewable power generation also takes part in grid stability. An environment as dynamic as this, demands active control mechanisms, which in turn require ICT for its management [35]. The power distribution grids located at the Low Voltage (LV) and Medium Voltage (MV) levels, are currently operated largely without ICT, whereas in contrast to the transmission grid, where generation and consumption is already optimized by Supervisory Control and Data Acquisition (SCADA) systems comprising the Ultra High Voltage (UHV) and High Voltage (HV) levels. However, in a smart grid this situation should change with increasing degree of automation, adding flexibility and greater integration of distributed generation by the implementation of a communication infrastructure for remote control.

According to the APG report [25], this paradigm change in the energy sector leads to increased intraday energy exchange and utility intervention continue to increase. Today, the power distribution grids operate increasingly at the limits of their capacity while bidirectional power flows are on the rise. The possibilities of decentralized generation, controllable loads and power storage provide flexibility for grid operators but also require ICT support. Furthermore, recent studies suggest that these underlying market functions can be attacked with new approaches that are enabled largely by the interconnectivity of distribution grid participants, as presented in [30].

Much like todays' generators located on the HV-level, future distributed generators must contribute control reserves to participate in supporting global and local stability measures. During a large-scale power outage basic operation and also a black-start, i.e., restart of a de-energized power grid, must still be possible even without ICT support. A backbone communication infrastructure for a redundant supply of the most important basic information would be an advantage here. From a functional and economical perspective, the fusion of different ICT structures such as smart metering and industrial control networks makes sense. However, such an approach involves risks concerning cyber-attacks [142].

Smart- and Micro-Grids as alternative Management Strategies

Because power grid management is nowadays organized in a centralized fashion [25] and high level failures directly impair the lower levels, alternative management concepts are being developed that are less dependent on the transmission grid. Ilo [78] introduces a new concept for the holistic view of a future power grid and considers all hierarchy levels. The "Supply Chain Net" states that one could upgrade the open-rings of distribution grids to micro-grids that balance and trade energy with neighbors under ICT management.

In a decentralized future, many distribution grids would remain connected to the transmission grid but should be able to operate autonomously. These micro-grids would be able to balance consumption and generation within their boundaries. Additional demand would be communicated to neighbors and the TSO's's. During times of insufficient local generation, power can be requested from alternative sources. Otherwise consumers have to be dropped in favor of grid stability and priority given to critical consumers such as water supply or public authorities. In case of a Europe-wide blackout, many microgrids could run autonomously for a period, helping restore a stable power grid. Decentralized generation may expedite the restoration process in this case. Still, water supply and emergency services could be covered with a small amount of load generation [44].

Kaufmann et al. [101] introduce new islanding capabilities of current distribution grids, so-called "operational-modes". They regulate the transition between grid-connected and island-mode in both directions. They also integrate decentralized energy storage in the operational management for enhanced management capabilities. Unlike a centralized approach, such decentralized approaches establish the possibility of coupling micro-grids on a superior authority, yet grants self-determination. Such a overlying authority negotiates capacity or energy exchange between micro-grids and other actors. The basic idea being that micro-grids are self-sufficient, however during times of under-supply, energy from outside the borders may be requested. Consequently, micro grids with excess energy offer it on the market. Although micro-grids require ICT, they should minimize communications to conserve bandwidth and minimize complexity. Such a system distributes intelligence and does not have a single point of failure, cf. Section 3.4.4. When a power grid collapse is imminent, micro-grids can save themselves in an islanding operations mode. Through load shedding and energy management it may then be possible to rebuild and resynchronize a power grid. However, such a decentralized architecture may host insufficient distributed generation in urban areas. Micro-grids may lead to increased resilience due to the continued supply of many autonomous sub-grids but also to increased control effort.

Smart grids with their extensive communication capabilities serve the optimization of power generation and consumption along side with smart metering and market services. One of the goals of smart grids is to maintain the current supply quality despite a high percentage of decentralized generation. Various influences come into play which on one hand take place in the electricity- and on the other hand in the communications-domain. The electricity grid already has significant dependencies to ICT for management purposes. Yet, decentralized generation requires additional control mechanisms. The ICT required for such a control effort faces security aspects such as malware based cyber-attacks. Both domains, power and communication, have a critical effect on the reliability of energy supply [36].

Smart grids utilize ICT to increase efficiency and reliability in the management of dynamic power consumption and generation. Although ICT allows for new capabilities, it also increases the attack surface of smart grids. This issue must be addressed in order to ensure the security of future energy grids. Khan et al. [103] and Yu et. al [186] provide a comprehensive survey of technologies, applications, case studies, architectures, and security issues [44]. According to Line et al. [115] the energy industry is traditionally well-prepared for threats such as physical damage, accidents, natural disasters, or equipment failure as long as they affect small, restricted areas. However, coordinated cyber-attacks can do significant coordinated damage, yet are still inadequately addressed, due to their low probability of occurrence. Using coordinated, distributed resources, cyber-attacks can theoretically target a sufficiently large number of critical equipment to originate cascading effects and eventually cause the system to collapse. Several incidents of attacks on the energy industry have been reported by the Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) [132] that demonstrate how organized groups with motivation, resources, and competence can cause serious damage. They report that about 60% of all cyber-attacks in the year 2013 targeted the energy sector, albeit not necessarily power control networks [48].

2.1.2 Heat Grid

We discuss heat grids as a subtopic, as our focus lies on the electricity grid and its corresponding ICT. However, heat grids rank among critical infrastructures [51], cf. Appendix A, thus failure in one can have serious implications on others, especially during cold seasons. Heat grids are typically localized water grids with a small geographic proliferation. Due to physical limitations such as pressure or thermal losses, inter-state heat grids are not feasible. However, they often supply metropolitan areas, making them a relevant player for a region. We consider heat grids among other supply grids in Section 2.1.4, where several grid types intersect for energy exchange. [100]

2.1.3 Gas Grid

Similar to Section 2.1.2 gas grids are also considered among critical infrastructures [51] as they are essential to deliver energy in the form of natural gas into households, industries, and businesses. Their geographical proliferation is huge compared to heat grids, because gas grids are often supplied through pipelines from other countries or continents. Yet, they do not reach the proliferation and density of the electricity grid. Furthermore, gas grids are also considered in Section 2.1.4. [100]

2.1.4 Energy Hubs

Energy hubs represent intersections where electricity, gas, heat and other forms of energy are redistributed. The basic idea is to rebalance overproduction in one energy-grid with transformation into another energy-form, e.g., decrease voltage-levels (electrical), or gas-pressure (chemical), by heating water (thermal). Other examples include burning gas (chemical) for electricity, or utilizing power-to-gas (chemical) to produce electricity. Such energy hubs can transform and balance several forms of energy and should, therefore, be considered critical nodes, in an ICT sense, due to their high level of interconnectivity. Kaufmann et al. [100] elaborate on several aspects of energy hubs and bring forward a simulation model. We consider energy hubs from an security point-of-view as an ideal attack point in hybrid grids, with a multitude of attack vectors.

2.2 Communication Networks for Smart Grids

Communication networks are already a fixed element of power grid optimization. Historically centralized control units managed by SCADA systems are characterized by one central decision-maker unit that monitors and optimizes all subjacent distribution nodes. This centralized entity has absolute authority and absolute knowledge over all processes [51]. If rendered inoperable, serious consequences for the entire system can be expected. Since large amounts of data must be transported over vast distances, real-time and non-real-time data depends on the availability of network bandwidth and integrity of the data. Therefore, measures to ensure minimum requirements for the continued operation must be taken. The following list provides oversight on how the communication in a smart grid should be organized [145]:

- *Minimum requirements:* A basic supply of power to the other critical infrastructures, e.g., water pumps, emergency services, authorities, and heat grids, even without ICT functionality, must be established in a rudimentary form. Therefore, it is necessary to operate basic functions of the power grid manually.
- *Backbone ICT supply:* Basic communication between sub-networks via a backbone infrastructure provide ICT supply to the LV transformers, leading to a reduced services capability.
- *Optimum operation:* Energy flow optimization, energy market, efficiency optimization, smart meter operation, among others can be provided when all basic services are functioning.

Several reference architectures for smart grid communication networks have been proposed so far. SGAM [21] offers a holistic approach that encompasses multiple dimensions ranging from the physical component to the business level. The BSI [61, 62] establishes a protection profile for a smart-meter gateway, for network segmentation, and data handling. NIST [174] outlines a generic model for security strategies and a centralized architecture with meta-level requirements. ENISA [49] offers security recommendations, outlines risks and challenges, and provides a knowledge inventory. SG^2 (Smart Grid Security Guidance) [112] uses a threat and risk analysis to explore the impacts of and countermeasures against cyber-attacks. Key results include a threat catalog and proposals for countermeasures with effective encryption and authentication measures. Furthermore, they find that while embedded security, e.g., Internet-of-Things (IoT), is still immature, reducing the attack surface may help prevent attacks. The results were combined into a holistic model that can be extended to future ICT functionalities. Khan et al. [103] provide a comprehensive survey of applicable technologies, architectures and security considerations with detailed methods. Akhtar et al. [3] survey wireless sensor networks and their power supply with regard to renewable resources, storage technologies, and wireless power transfer [44].

2.2.1 Smart Grid Architecture Model

Our simulation models retain close proximity to the established SGAM model [21] for consistency in our topological approach.

Furthermore, our model, cf. Section 3.4, operates between the field, station, and operation zones. We also consider those zones part of the control network which should be segmented securely for scrutiny reasons. However, we do consider the enterprise zone (outside our control network) in our countermeasures, cf. Section 10, yet do not extend our simulations there. The enterprise zone represents the enterprise networks which connects hosts via Remote Terminal Units (RTU) to the control network. We simulate in the control network and define it as the system boundary. Furthermore, we do not consider the market or the process zone in this work. Concerning domains in SGAM, we span our simulations from the customer premise to the transmission domain. Furthermore, our model operates between the component and information layers.

2.2.2 Wide Area Monitoring Systems

The North American Synchro-Phasor Initiative (NASPI) [136] and Kanabar et al. [93] present recent advancements using PMU's as a base technology for monitoring the power grid. The measurements help achieve situational awareness and serve as input for control functions. Most Wide Area Monitoring Systems (WAMS) are organized in hierarchical architectures with a

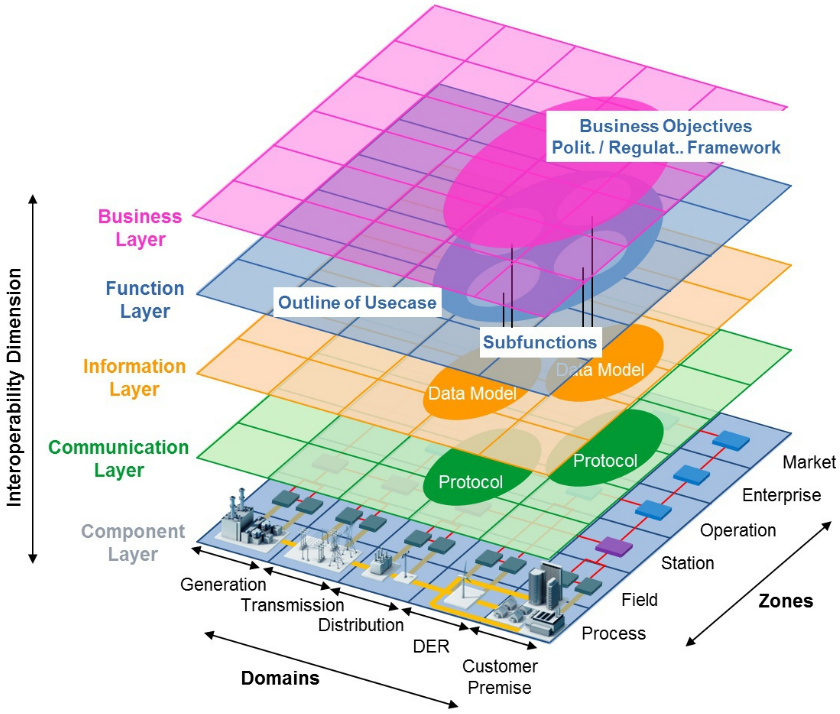


Figure 3: Smart grid architecture model, as presented in [21]

control center. According to Zhang et. al [188], WAMS manage the data exchange among control centers and state estimators, which apply statistical methods to make decisions based on the collected data. As dynamic renewable and distributed energy sources become more widespread, monitoring functions utilizing WAMS become increasingly important [44].

2.2.3 Advanced Metering Infrastructure

According to Dan et al. [32], Advanced Metering Infrastructures (AMI) collect data on power consumption and transmit them to the utility company. This smart metering approach allows for automated billing and enhanced observability in the power distribution grid. Technologies used to achieve smart metering include Power Line Communication (PLC), dedicated wire, public mobile carrier, and wireless networks. Bou-Harp et al. [17], Yan et al. [185] and Khan et al. [103] list technologies such as Zigbee, WiMAX, Wifi, and GSM as possible means of supporting smart metering via wireless solutions. As mentioned earlier, the BSI [61, 62] defines security considerations for AMI that use a gateway connecting smart meters, concentrators, home appliances, and also proposes specific network protocols. We elaborate on these communication technologies below [44].

Table 2: ICT requirements for smart grid applications [68]

Application	Bandwidth [kBps]	Latency [s]
Substation automation	9-56	0.015 - 0.2
Overhead transmission - Line monitoring	9-56	0.015 - 0.2
Home energy management	9-56	0.3 - 2
Automated metering	10-100	2
Wide area monitoring	600-1500	0.015 - 0.2
Demand response	14-100	0.5 - >60
Outage management	56	0.02 - 0.2
Distribution management	9-100	0.1 - 2
Distributed energy resources - and energy storage	9-56	0.3 - 2

2.2.4 Technology Requirements for Smart Grid Applications

This section discusses the state of the art and the requirements on ICT infrastructures for smart grids, cf. Table 2.

There are a number of technologies available for the use with smart grid communications. We elaborate on wired and wireless technologies in the following sections and differentiate them by scope, i.e., Wide Area Network (WAN), Metropolitan Area Network (MAN), Neighborhood Area Network (NAN), and Home Area Network (HAN).

2.2.5 Wired Link Technologies

According to [4, 75, 79, 111, 117, 164] the technologies Digital Subscriber Line (DSL), Coaxial, PLC, Passive Optical Network (PON), Synchronous Optical Network (SONET), Ethernet, and Optical Transport Network (OTN), are generally used in wire-line smart grid applications. Table 3 lists those technologies, the relevant standards, their areal coverage, and scope.

2.2.6 Wireless Link Technologies

According to [1, 4, 54, 55, 75, 117] the technologies WiFi, Worldwide Interoperability for Microwave Access (WiMax), Mobile Carrier, Satellite, Wireless Personal Area Network (WPAN), and Z-Wave are generally used in wireless smart grid applications. Table 4 lists those technologies, relevant standards, their areal coverage, and scope.

Table 3: Wired communication technologies [4, 75, 79, 111, 117, 164]

Technology	Relevant standards	Areal coverage [km]	Scope
DSL	ADSL, VDSL	< 7	HAN
Coaxial	DOCSIS	< 10	HAN, MAN
PLC	IEEE 1901, IEC 61334 ISO/IEC 14908-3	< 150	NAN, MAN, WAN
PON	IEEE 1901, ITU-T IEEE 802.3ah (EPON)	< 60	HAN, MAN, WAN
SONET	SDH: ITU, G.707, G.783, G.803, T1.105	< 20	MAN, WAN
OTN	ITU-T	< 10	HAN, MAN
Ethernet	IEEE 802.3	< 70	HAN, MAN, WAN

Table 4: Wireless communication technologies [1, 4, 54, 55, 75, 91, 117]

Technology	Relevant standards	Areal coverage [km]	Scope
WiFi	IEEE 802.11	< 1	HAN, NAN
WiMax	IEEE 802.16	5 - 30	MAN, WAN
M-Bus	EN13757-3	< 1	MAN, WAN
Mobile carrier	GSM, Edge, LTE, UMTS, HSPA	< 30	MAN, WAN
Satellite	SDR, S-UMTS, GMR	Beam dependent	MAN, WAN
WPAN	IEEE 802.15.4, Zig-Bee, Bluetooth	< 0.07	HAN, NAN
Z-Wave	ITU-T G.9959	< 0.07	HAN, NAN

2.2.7 Differences between Smart Grid Networks and Internet

Due to the critical role that electricity grids play in a nations' society and the increasing interconnectivity of smart grid devices via ICT, incentives and opportunities to attack smart grids are increasing. Furthermore, the entire range of sophisticated malware known from the Internet is increasingly suited for deployment against smart grids. We describe the basic smart grid environment and identify essential properties that differentiate it from classical Internet communication. This section defines the capabilities and characteristics of attackers, and Section 4.3 elaborates on security assumptions on which our proposed attack life-cycle model relies upon. These discussions establish the general operational environment of this work [48].

Smart grid ICT infrastructures differ in several aspects from classical Internet communication. Smart grid communication is mainly based on Machine-to-Machine (M2M) communication, which makes it more predictable than traditional Internet traffic, i.e., Human-to-Human (H2H) or Human-to-Machine (H2M) [174,189]. Predictability simplifies anomaly detection, but some challenges remain, depending on the choice of the observation points and the protocols in use [48].

Furthermore, security and privacy concerns upon smart grid applications may lead to conflicting goals. For instance, smart meters collect data on energy-consumption, therefore, could be abused to monitor user-behavior which raises privacy concerns. Such data could be merged with personal information collected about each user on the Internet to yield even more detailed data. Many smart meter types are also capable of remote disconnection. Such features raise security concerns, because they can be abused to disconnect many households at once [189], leading to imbalances in the power grid. Based on [189] and [174] we identify the following characteristics that distinguish smart grids from Internet communication: [48]

- The predominance of M2M communication, instead of H2H or H2M communication.
- Homogeneity (monocultures) in the choice of devices.
- Differences in network requirements across several device types (smart meters, gateways, sensors, actuators, etc.).
- Physical access to field devices by non-trusted parties.
- Huge planned life span of installations in the field, e.g., utility companies plan for smart meters to be operational for more than 10 years.
- Pre-authorization of field devices to local servers, e.g., pre-configured certificates or authentication tokens that are stored in field devices.
- A need for remote monitoring, maintenance, and updates. In particular the requirement for remote upgrade support.

2.3 Security of Critical Infrastructure Networks

This Section discusses the state of the art on security topics concerning critical infrastructures. Additional details are included in Section 4.

The national strategy for the protection of critical infrastructures (KRITIS) [60] by the German Federal Ministry of the Interior states that society is vulnerable to a number of events listed in table 5. It considers the resilience in critical infrastructures as a means to reduce the impact of technical failure and natural events to the power grid [44]. However, resilience against

Table 5: Vulnerabilities and risks for critical infrastructures, according to [60]

Natural events	Technical / human failure	Terrorism & war
Extreme weather	System failure	Terrorism
Forest fires	Negligence	Sabotage
Seismic event	Accidents / emergencies	Other crime
Epidemics / pandemics	Failure in organization	Civil war
Cosmic events		War

failure does not prohibit attackers from deliberately targeting those resilient structures.

These infrastructures have been listed by the European Commission [51], and deemed critical to the continued functioning of society, cf. Appendix A for a poster of critical infrastructures with examples: [44]

- Energy grids and supply
- ICT connectivity
- Water supply and wastewater disposal
- Food supply
- Medical care
- Emergency services
- Public order
- Financial services
- Transportation

Furthermore, KRITIS [60] states that these infrastructures are susceptible to a number of events. They may be disruptive to any critical infrastructure, thus, may restrict the everyday lives of the general population.

Table 5 lists the category "terrorism and war" which includes malicious attacks on critical infrastructures. Some cyber-attacks can even be conducted anonymously via cyber-attacks due to increasing interconnections between networks.

As mentioned in Section 1.7 different malware types are capable of exploiting a vulnerability in networked devices. For instance, some malware can extend their functionality by downloading modules. Furthermore, they exist with several different characteristics, which are discussed in detail in Section 8.1.

Since some malware types communicate with each other, forming a botnet, i.e., a remote Command-and-Control (C&C) infrastructure, with a botmaster, they are capable of perform flexible commands, in order to react to the environment or adjust to different attack goals. Extortion schemes, Distributed Denial-of-Service (DDoS) attacks, espionage campaigns, repurposing attacks, and lately also cyber-physical attacks are examples of complex attacks on networks that can be triggered. Section 4.4.6 and Table 25 elaborate on them [48].

Modern cyber-attacks, however, are increasingly conducted by so-called Advanced Persistent Threats (APTs). These groups often utilize obfuscation techniques in order to remain concealed for long periods. Their attacks seem to be increasing the complexity, versatility and potential damage. According to Thonnard et al. [176], most APTss utilize zero-days, i.e., widely unknown vulnerabilities. This further lowers the chance that affected defenders can discover an attack. Therefore, attackers are increasingly capable to further obfuscate operations and utilize attack methods that are highly challenging to defend against [48].

2.3.1 Cyber Kill Chains and Attack Models

We considered several attack models proposed by existing work, and have based our generic life-cycle model, presented in Section 4.4, on them. Lockheed Martin [118] introduced the cyber-kill-chain as part of a framework for intelligence-driven defense to prevent network intrusions. It comprises the following stages: reconnaissance, weaponization, delivery, exploitation, installation, control, and action. The SPARKS project [128] proposed a similar linear kill chain composed of the following successive stages: Intrusion, installation, lateral movement, exploitation, pivoting to the control network, deployment, and attack. Another linear approach was taken by Matrosov et al. [123] using a client-server model with focus on the client side. With social engineering as the intrusion vector, this approach cites local exploits and malicious infection as methods of gaining persistent access. Schneier [154] introduced the concept of "attack trees" that branch off at each decision an attacker makes. This formal model is well known for its versatility, as it is able to describe any attack according to its scalability. All four approaches provide a clear sequence of events that occur during a cyber-attack. We considered these models in developing our own generic approach by reusing parts of their main concepts [48].

In addition, Gollmann et al. [67] have discussed cyber-attacks in which each stage is connected to the others. These stages include access, discovery, control, damage, and cleanup. Rather than following a fixed sequence, they argue that malware can move from any stage to another at any given time. We provide details on each stage in our model and extend this approach in several aspects. We also elaborate on logical sequences these stages can follow under different circumstances. Li et al. [114] discussed several characteristics in target-finding, propagation and transmission schemes that we also took into account in our work. Trullols et al. [177] have researched large-scale vehicular networks and introduce attack stages and target discovery schemes that complement those scanning techniques we include in our model [48].

Since smart grids are a critical infrastructure, they deserve progressive protection from cyber-attacks, e.g., security updates should be deployed in a timely manner as unpatched vulnerabilities increase their attack surface. Vulnerabilities based on shortcomings in the hardware may even persist for prolonged periods, as replacing widely deployed hardware is accompanied by financial and staffing issues. This is why, aside from APTs, malware that uses readily available technologies in a recombining fashion is a realistic threat to modern power grid control networks. The attacker can choose among a multitude of targets in a large communication infrastructure and the electricity grid. Targeted equipment includes power switches, transformer stations, and field devices, e.g., smart meters or PMUs. [18, 48]

2.3.2 Threats to Critical Infrastructures

According to the National Institute of Standards and Technology (NIST) [174], future smart grids require security measures on the basis of architectural design (security by design) among other means against an array of typical adversaries for ICT, which include: [44]

- *Nation States* are organized and well financed groups that attack critical infrastructures with highly evolved cyber-weapons.
- *Hackers* attack networks to exploit vulnerabilities. Skilled hackers may be capable of sophisticated attacks, albeit not at a nation state scale.
- *Terrorists* mainly follow political agendas, often without consideration of collateral damage.
- *Organized Crime* coordinate well financed activities.
- *Other Crime*, another facet of crime, not well organized, but dangerous to inattentive victims.
- *Industrial Competitors* may be engaged in the illegal gathering of information by means of espionage.
- *Disgruntled Employees* may represent an inside threat. They often have many options available to deal damage to critical infrastructures.
- *Careless Employees* pose a threat through a lack of training, concern or attentiveness.

Threats such as malicious nation states, organized crime or dedicated disgruntled employees will remain a threat even with high security measures implemented, as they have the resources to develop attacks that can infect critical nodes on high levels of the hierarchy. However, decentralized systems, devision of powers, and strict policies may help to mitigate some incidents arising from such threats. The main entry points for less equipped attackers are assumed to be located at the lower ICT levels which have low physical security and are publicly accessible. If such devices are connected in a mesh network, malware attacks may propagate quickly from the power grid control network to other grids, e.g., water- or gas-grids, cf. Figure 4. We assume that smart grid communication uses secure protocols to establish confidentiality, integrity, and authenticity. However, new security vulnerabilities in soft- and hardware arise every day that can be exploited. Therefore, the ICT architecture should be designed to prevent or slow the spread of attacks [44].

Considering that today's power transmission grid is controlled by legacy SCADA systems, Igure et al. [77] state that after the year 2001 about 70% of all incidents concerning SCADA systems are attacks from outside the network. However, the future smart grid deploys ICT on a wide scale and mainly on the power distribution level enabling new attack vectors, when protected insufficiently. Attacks of varying motivation are to be expected, whereby future smart grid operators must implement security means, at the very least, to repel adversaries with limited skills and resources. They may be located at the lower ICT levels, thus, publicly accessible, connected via smart meters, smart home devices, or HAN. Therefore, they should not be considered trustworthy. ICT devices installed at the MV-level can be considered more secure, because they can be protected inside transformer rooms. Yet, they do not protect against dedicated attackers [44].

Nevertheless, ICT must mitigate malware propagation by implementing vertical and horizontal security:

- *Vertical security:* Measures against propagation to higher layers in the hierarchy, e.g., smart meter to concentrator.
- *Horizontal security:* Measures against lateral movement, containing the spread of malware between entities on the same level.

Traditional cyber-attacks target individuals, businesses, intelligence services, and military targets. However, the latest successful attack on the Ukrainian power grid [113] manifests a new impact level that affects the civil population. Attackers managed to infiltrate utility companies and cause damage to power switching equipment that led to wide ranging blackouts. We discuss the methods used by the attackers, cf. Section 4.4.1, which could in part have been prevented with basic defense measures, cf. Section 4.3. Therefore, Line et al. [115] consider Cyber-Situational-Awareness (CSA) to be a future cor-

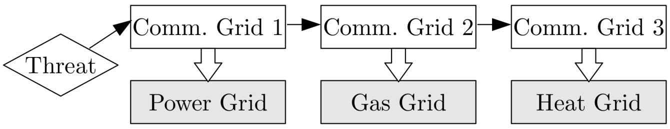


Figure 4: Incident propagation, as presented in [44]

nerstone in protection against intruders. CSA attempts to mitigate threats in the propagation phase, before attacks are conducted successfully. The goal is the awareness and comprehension of: [48]

- The current situation inside the network.
- The situational evolution during an attack.
- The causes and implications of the current situation.
- The quality of the collected information.
- The impact of attacks on critical equipment.
- The attackers behavior before, during, and after an event.
- The possible future developments and recovery plans.

2.3.3 Incident Propagation and Cascading Effects

This section discusses failure incidents that can be abused for attack vectors when triggered intentionally. Christiner [25] describes the propagation of a broadcast message from the control system of a gas grid into the power transmission grid SCADA system due to a misconfiguration. The message proliferated, similar to a DDoS attack. The SCADA-system was not usable until the rouge broadcast messages had faded out. During this incident, the grid operators were unable to monitor or control the power transmission grid. However, no blackouts occurred thanks to the manual control of TSO's operators. This real world example shows that incidents in one infrastructure can indeed spread to other infrastructures. However, in future smart grids the number of networked nodes will be much higher, complexity increased substantially, rendering manual override more difficult [44].

Figure 4 shows incident propagating across different networks that are used to control physically separate grid types.

Another example involves a cascade of overloaded power lines [180], which resulted in the division of the European power transmission grid into three islands, each with a different frequency. Enough reserve generation power was available to maintain the stability of each island and to resynchronize them. However, this example shows that incidents can propagate across an entire continent within a short period.

Recovering from such disruptive incidents poses another problem. Since few black-start-capable power stations are available, ENTSO-E [53] has developed extensive restoration guidelines for power grid operators. When rebuilding a collapsed grid, its components, e.g., cables, transformers, and generators, must be re-energized and non-self-sufficient power plants can only be connected after stable grid operation is established. Only then, it is possible to reconnect loads along with more generation units. Bruno et al. [18] outlines this lengthy process of restarting a collapsed power grid in detail. Klick et al. [105] present further research on vulnerable industrial controllers that could lead to critical vulnerabilities in widely used industrial equipment that may be involved in such incidents. According to Burke [19], cyber-attacks on power grid controls already occur frequently, unbeknownst to the public.

2.3.4 Monetary and Public Cost after a Blackout

This section discusses the value of energy supply-security and the danger of blackouts to economic regions. The Johannes Kepler University [50] assembled a publicly available simulation environment, which can be used to simulate estimations on economic losses due to electricity blackouts. This tool covers all European countries. We provide one example: I.e., a blackout over the duration of one business-day in Vienna, that causes an estimated financial damage of 230 million Euro [50]. Reichl et al. [146] elaborate on this with extended simulations, which conclude that a twelve hour blackout across the entire country of Austria leads to total financial losses of 477.7 Million Euro.

We note that the monetary losses do not increase linearly with the duration of a blackout. Instead, prolonged exposure to a collapsed power grid quickly leads to a number of follow up-effects collapsing other critical infrastructures along with social cohesion, cf. Appendix A.

3 Network Architecture Model

Notice of adoption from previous publications:
Parts of the content in this chapter have been previously published in [44]. We include basic assumptions on the electricity grid hierarchy and corresponding network topologies, which outline differences between the energy- and the ICT domain. These variants range from fully centralized to fully decentralized types, and include hybrid concepts. The main contribution of the author was an extensive analysis of network topologies suitable for smart grid networks. The co-authors contribute valuable additional information in discussions. These topology models were also presented in [37, 39].

This section discusses the abstraction models for network technologies and builds the theoretical basis for further simulations. We include topological, technological, statistical, and malware abstractions which are based on confirmed real-world examples.

3.1 Methodology

We investigate research question 1, cf. Section 1.3, with theoretical considerations and several established smart grid models. This extensive investigation provides the basis for our network topologies suitable for smart grid networks. We then used real power grid data to support our theoretical models that are used in our simulations. The following sections presuppose some results that could be relocated to the results section. However, we use them to populate our network topologies for the simulation model, thus, already include them in this earlier section, to use them as the basis for our models.

3.2 Architecture Hierarchy Levels

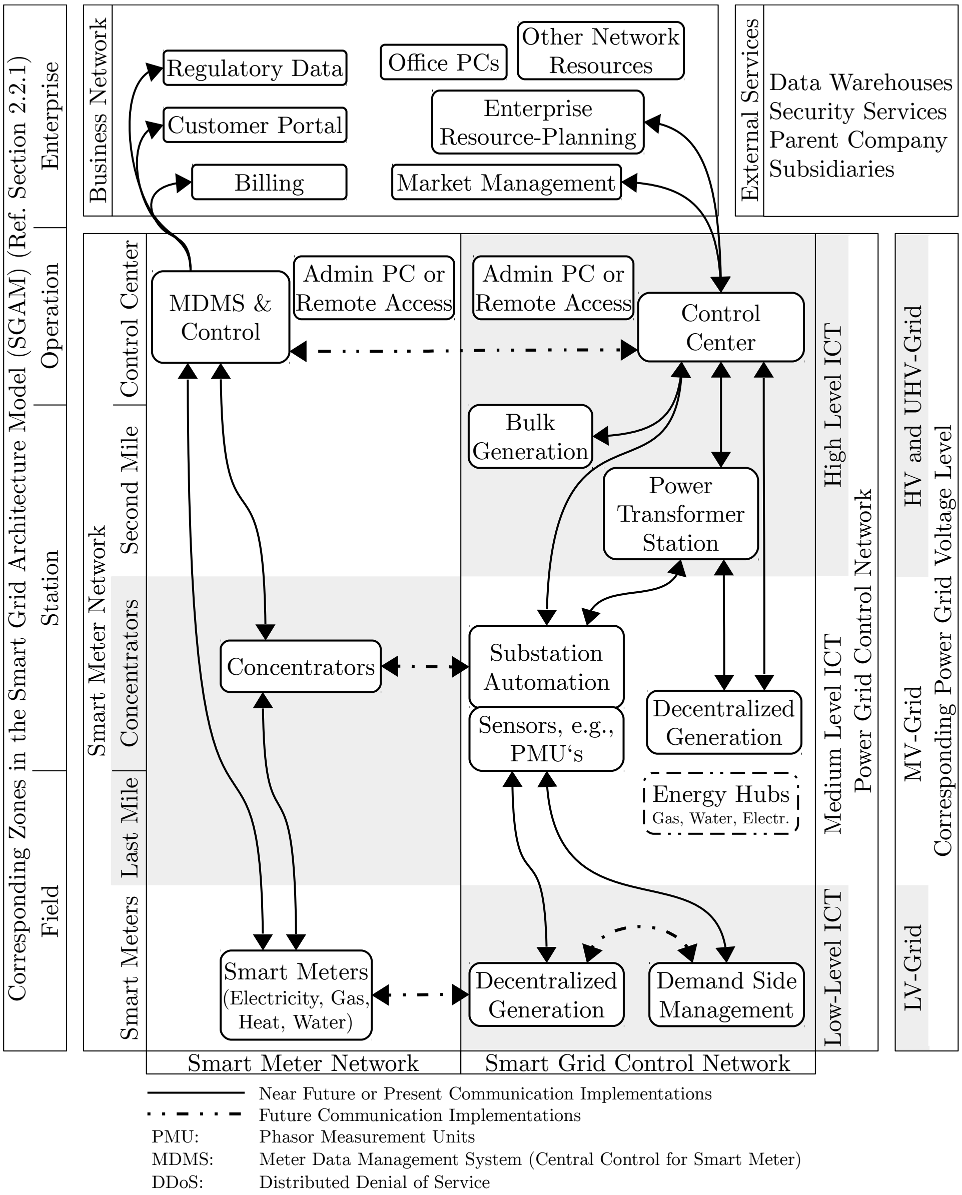
We setup our hierarchy model in line with SGAM, cf. Section 2.2.1 and Figure 5. SGAM recognizes that the electricity grid hierarchy coexists with the ICT topology and separates them into domains and zones. It spans across all levels of the power grid, including the energy market and political stakeholders. However, we model our simulation based on one single dedicated network, responsible for both smart metering and power grid control across all levels of the hierarchy, representing our smart grid. Today, these systems are separate, however, in the future this may not be the case, especially when smart meter systems mature reliable switching capabilities. The ICT hierarchy is based on the electricity grid hierarchy and includes nodes for centralized control functions and data aggregation, nodes for local intelligence, and nodes for distributed control. Section 3.3 elaborates on each node type in detail. The enterprise network is not included in our simulations, because we assume effective network segmentation measures, as discussed in the defense measures list in Section 10. We, therefore, exclude external attack vectors, e.g., through spear phishing attacks, waterholing attacks, and lateral movement sourced from the enterprise network. These attacks can result in direct infection of critical remote terminals inside the control network, thus, they are capable of immediate and extensive damage, when not segmented properly.

3.3 Node Types and Communication patterns

We elaborate on details of the node types and connection patterns that are used in our model. For this, we define the node types and their communication patterns, based on device types, expected in smart grid control systems.

- *High-Level-nodes (HL-nodes)* are regional control nodes, that are typically located in large power distribution stations. They aggregate data from the lower levels and forward it to the control center, or represent the control center itself. These nodes may execute control functions over regional power switching equipment, e.g., distribution transformers. Therefore, they represent critical nodes which we refer to as high level ICT nodes, in short *HL-ICT* or *HL-nodes.*
- *Medium-Level-nodes (ML-nodes)* act as local control nodes. They aggregate data from the lower levels, thus, act as a middle man, and include, e.g., PMU data or other local sensors. They can also act as local firewalls or event loggers, although we do not attribute them these security functions in our simulation model. The ML-nodes are typically located in local transformer stations, thus, behind locked doors with some physical security. However, they are considered field nodes and are, therefore, of limited trustworthiness. We name them medium level ICT nodes, in short *ML-ICT* or *ML-nodes.* They can aggregate data from local PMU's or other sensors.
- *Low-Level-nodes (LL-nodes)* consist of several smart meters and a building energy manager unit, as elaborated in Section 3.5.2. They represent the main power connection of, e.g., houses and distributed energy sources. We name them low level ICT nodes, in short *LL-ICT* or *LL-nodes.*

Table 6 summarizes the communication patterns used in our simulations. Different types of nodes have a distinct communication pattern that depends on their function. Therefore, a great number of LL-nodes communicate less



Near Future or Present Communication Implementations
Future Communication Implementations
PMU: Phasor Measurement Units
MDMS: Meter Data Management System (Central Control for Smart Meter)
DDoS: Distributed Denial of Service

Figure 5: Hierarchy of power grid control network, smart meter network, business network and external services

frequently to their master node, the ML-node, than all ML-nodes would communicate with their master node, the HL-node.

Table 6: Communication traffic between nodes

Communication Type	Interval [s]	Message size [kB]	Reference
ML-nodes to HL-node	1	100	[5, 6]
LL-nodes to ML-nodes	60	100	[5, 6, 120]

We illustrate legitimate and malicious communication in Figure 6 across different networks. We define our communication model such that one node can only have one network interface per network and we exclude redundant network interfaces or looped connections. Furthermore, our malware model, cf. Section 4, defines that those nodes with more than one interface can be infected from any interface, immediately setting all remaining interfaces to the infected status. Additionally, legitimate communication is defined as LL-nodes communicating with ML-nodes, and ML-nodes communicating with HL-nodes. All nodes may initiate the communication.

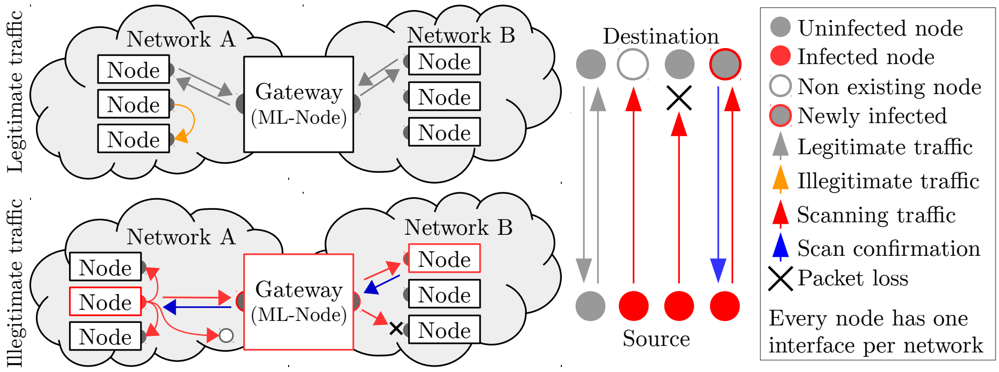


Figure 6: Communication traffic and patterns, as presented in [47]

3.4 Theoretical Topology Models for the ICT Network

This section considers our topological abstractions. We derive four basic topologies [44] for our simulation environment. The results correspond to research question 1, cf. Section 1.2, that investigates the most promising features of security by design against malware propagation.

Figure 7 and Table 7 present a comparison of the ICT topologies according to power grid hierarchy type. The centralized, cell, and mesh topologies, cf. Figure 7-a, b, c, differ only at the power distribution level, i.e., LL-ICT and ML ICT. These topologies assume that legacy SCADA systems remain in service at the HL-ICT level. The fully decentralized approach, cf. Figure 7-d,

Table 7: Control mode comparison at all hierarchy levels, as presented in [44]

	Corresponding Power Grid Hierarchy			
	HV	MV	LV	HA
	ICT Hierarchy Level			
Control mode	HL-ICT	ML-ICT	LL-ICT	
a. Centralized topology	central	central	central	central
b. Cell topology	central	mesh	central	central
c. Mesh topology	central	mesh	mesh	mesh
d. Decentralized topology	mesh	mesh	mesh	mesh

represents an exception to the legacy SCADA regime, yet, requires upgrading them to mesh network capability. Generally, only necessary and predefined data should be communicated [44].

Silva [161] concludes that centralized topologies are beneficial over distributed mesh networks with regards to coverage, capacity, reliability, and cost. However, mesh networks transfer data hop-by-hop, which entails redundant paths, adding resilience, yet, resulting in increased protocol overhead and latencies. While centralized topologies are best for scenarios without security threats, mesh-based resilience becomes a valuable feature if ICT systems are threatened. Khan et al. [103] provide a survey of technologies for smart metering discussing smart grid communications including PLC, dedicated wires, public mobile carriers, or wireless networks. Cognitive radio in particular can be optimized through the utilization of many spectra [44,103].

We define six indicators used to compare our topologies, namely: [44]

- *Resource Control:* How well is the network topology suited to achieve situational awareness about the processes in the power grid? How effective is it in managing data, self organization and optimizing resources?
- *Security:* How well is the topology suited to mitigate cyber-attacks inside networks and across neighboring networks?
- *Resilience:* How well is the topology suited to mitigate failures of ICT components?
- *Quality of Service:* How does the topology influence communication quality in terms of protocol overhead and latency?
- *Compatibility:* How well is a topology suited to interface with legacy systems? Will an upgrade be necessary?
- *Cost:* What are the estimated financial (qualitative) implications for upgrading different topology types?

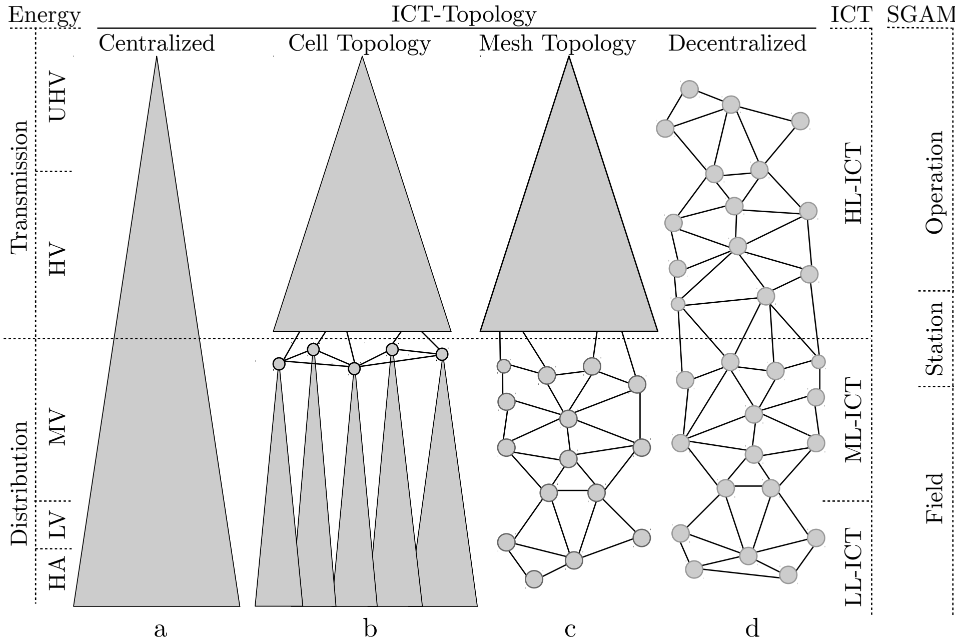


Figure 7: Comparison of topologies, as presented in [44]

3.4.1 Centralized Topology

Fully centralized topologies, cf. Figure 7-a and 8, collect all data in a single control center. There may be ML-nodes that distribute and forward data without decision power. When decisions are made in the control center, all commands are propagated across all levels, allowing situational awareness and the optimization of resources. However, as shown by ENISA [49], Kammerstetter et al. [90], Kupzog [110], Shin et. al. [157] and Van de Vyver et. al. [181], a centralized topology causes high latencies in data transmission, low flexibility, low resilience, congestion situations, and has a single point of failure [44].

Although redundant structures can mitigate errors in a single control center, planned attacks are not easily overcome by a backup system. *Architecture-based security*, i.e., security by design, must be considered during planning, in construction and deployment, rather than being retrofitted. A vulnerability at such a high level can lead to catastrophic failures. Yet a fully centralized architecture can provide some level of protection against malware, i.e., spreading horizontally is not possible, as nodes are connected only to higher levels. Malware may propagate vertically, but higher layers are usually better protected, e.g. through firewalls, segmentation, visualization, and physically secured buildings. Furthermore, the heterogeneity of components on differ-

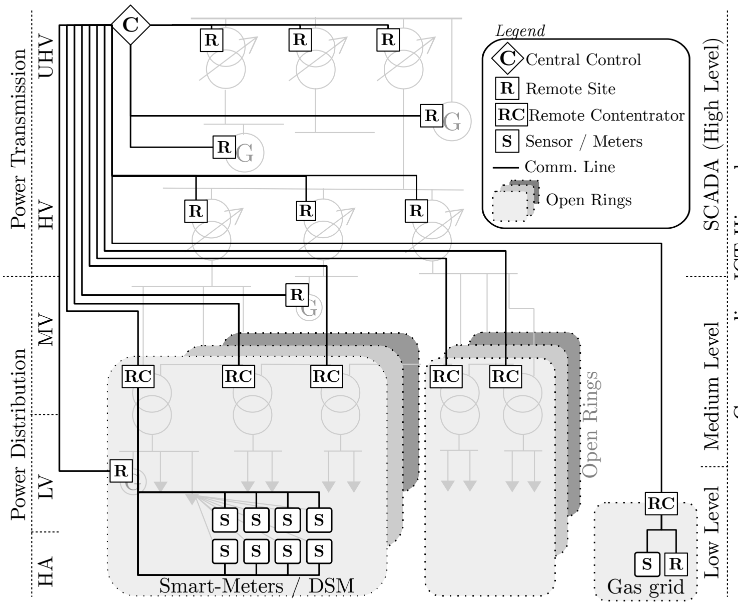


Figure 8: Smart grid with a centralized topology, as presented in [44]

ent levels of the hierarchy may foster resilience. Worth mentioning in this context is the potential threat resulting from the reuse of hardware, software, and design in various products from a single vendor. Primarily aimed at cost reduction, identical software or hardware at different levels of the hierarchy can enable malware propagation [44].

Centralized ICT may be practicable within the power transmission grid with few nodes. However, the number of nodes multiply rapidly when LL-nodes such as households with smart devices and distributed generation are taken into consideration. Furthermore, LL-nodes are not trustworthy, thus, they require special protection. It may not be feasible to control them via the legacy approach. The central control node would, in this case, become even more critical, thus, the main factor of all costs is concentrated there. Although this approach is less costly than it would be to protect many nodes in a distributed network [44].

In summary, the *benefits* of centralized topologies are [44]:

- *Resource Control:* Central data collection and control allow overall situational awareness that can be used toward resource optimization.
- *Security:* Physical access is controlled at the ML-ICT level, and lateral movement is impossible because of the hierarchical structure.
- *Compatibility:* Legacy SCADA can be integrated directly.

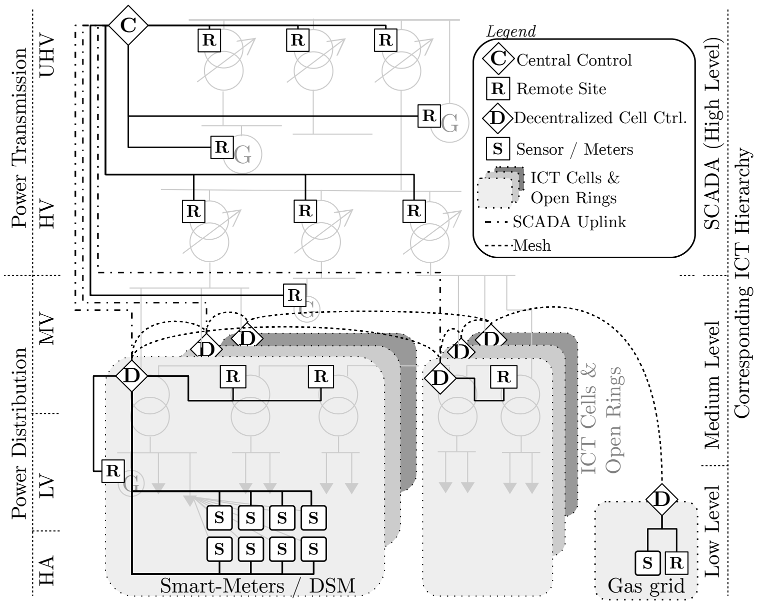


Figure 9: Smart grid in cell topology, as presented in [44]

- *Cost:* Higher layers with a small number of devices require expensive upgrades, while lower layers can function without local intelligence, overall keeping costs on a low level.

The *drawbacks* of centralized topologies are [44]:

- *Resilience:* LL-nodes fully depend on the HL-ICT. Redundancies of the HL-ICT may increase resilience against failure, but not necessarily against attacks.
- *QoS:* Excessive communication demands and long distances cause latency between nodes.
- *Security:* Malware may propagate vertically because there is no local control unit for analysis. However, central control is usually better protected than are distributed units.

3.4.2 Cell Topology

This section discusses a topology with designated cells in the ICT domain that match the electrical micro-grid concept, cf. Figures 7-b and 9. These cells are controlled by a decentralized agent called a cell controller as in the proposal from Kupzog [110]. This cell controller is located above the MV transformer, acting as a master node, data hub and local intelligence [44].

The transmission grid remains under the control of the legacy SCADA system, which connects the ML-nodes with the HL-node via dedicated uplinks. These uplinks provide information to SCADA systems in the interest of achieving situational awareness. Each cell acts autonomously, independent of SCADA control, and may also exchange information with its neighbors on the same hierarchy level. Therefore, SCADA manages the power transmission grid, while decentralized cell controllers, i.e., ML-nodes manage their subjacent local grids. Each cell consists of several MV transformers that are clustered under and controlled by one cell controller. The LL-nodes collect data for the cell controller. Smart metering and other services such as demand side management are controlled by the cell controller, which can act as a virtual power plant. For security reasons such services should not circumvent the cell controller, e.g., as would Internet based virtual power plants, as they converge data from lower levels and aggregate it for SCADA, other WAN entities and neighboring cells [44].

The cell controller's functions resemble those of the BSI gateways specified in [61]. These act as firewalls, segmenting networks and preventing communication among smart meters. They are usually located inside locked buildings and are physically more secure than smart meters. Local control makes the spreading of malware unlikely. As physical security is difficult to accomplish for LL-ICT, ML-ICT have to deal with compromised LL-nodes [44].

Customer data or control signals may be sent to recipients outside the cell controller only in aggregated form. Additionally, ML-nodes establish a mesh network in order to add resilience in case of a high level failure. Because they represent intermediate local control entities, cell controllers must be well protected against malware infections. Anomaly detection may be employed to preventatively warn neighbor cells and restrict communication. Measurement inputs are divided into *critical values* for stability such as;

- voltage (3 phase),
- current (3 phase and direction),
- frequency and phase angle,

and *non-critical values* for market signals concerning demand-side management;

- ΔP (Active power that is converted in mechanical work),
- ΔQ (Apparent power needed for magnetizing, e.g., transformers),
- smart meter data.

whereas critical values should have priority as discussed in the traffic light concept in [12, 44].

Shin et al. [157] argue that the most cost-effective approach to implementing smart grids is to utilize middleware in commercial communication infrastructures, e.g., GSM networks or Internet cable. However, sending control commands over shared networks opens new attack vectors. We argue that security concerns should restrict control functions to use dedicated networks. Shared networks may, in the worst case, be used on the lower levels where physical security is impossible to achieve. However, the ML-ICT and above must be able to provide security functions such as anomaly detection or firewall functionality, which represent the greatest cost factor in this topology type, thus should operate with their own dedicated network [44].

In summary, the *benefits* of the cell topology are: [44]

- *Resource Control:* Situational awareness can be established and resources optimized more easily than in fully decentralized or meshed environments due to the cell's hierarchical structure.
- *Security:* Physically secured cell controllers can control malware propagation. Restricting communications among cells and toward SCADA systems providing additional security.
- *QoS:* Local control minimizes data exchange and solves congestion issues.
- *Resilience:* Single cells may not be resilient to failure, however meshing cell controllers adds resilience, where the failure of one node does not endanger others.
- *Compatibility:* SCADA can be integrated into cells.

The *drawbacks* of cells are: [44]

- *Resource Control:* It is more difficult to establish situational awareness and optimize resources than in fully centralized environments. If the electrical topology changes, the renegotiation of ICT control is more complex.
- *Cost:* The highest costs occur at the cell controller level, which have to operate as autonomous entities with numerous functions. Higher layers need not implement extensive security measures.

3.4.3 Mesh Topology

The mesh topology, cf. Figure 7-c and 10, differs from the cell topology because all ML- and LL-nodes are meshed into a dynamic cluster at the distribution level that may form links across different types of grids, e.g. electricity, water or gas. Cell controllers are located on top of MV-transformers

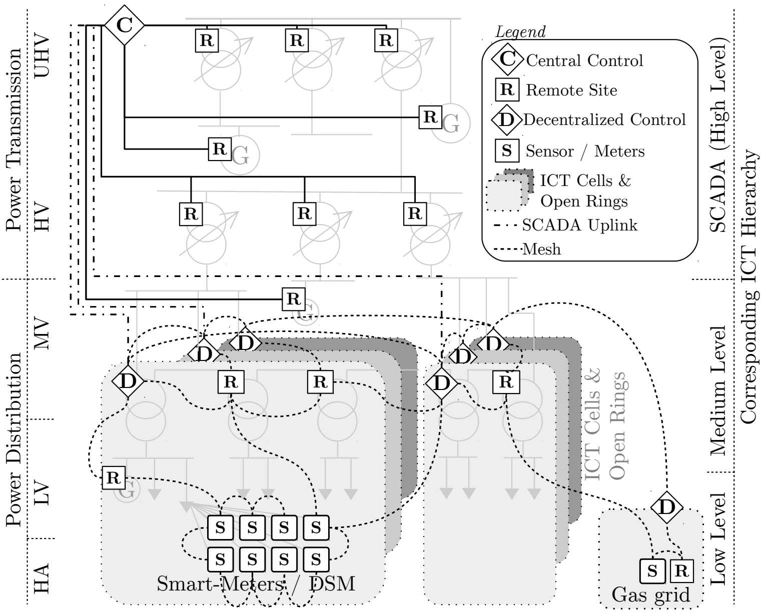


Figure 10: Smart grid in mesh topology, as presented in [44]

and organize local control as discussed in Section 3.4.2 [44].

In this topology, the SCADA system remains unchanged and communication at the lower levels occurs via a mesh network with local control units. These decentralized control units are under the control of the DSO and provide communication uplinks to LL-ICT. Meshed devices can form a mesh network across other grid types, circumventing some local control. However, as put forward by Christiner [25] and mentioned in Section 2.3.3, broadcast messages can propagate across vast distances and cause problems for other grid providers, when network segmentation is faulty. A future smart grid must be able to mitigate such misconfiguration, that pose a realistic threat [44].

Mesh structures are inherently more resilient to failure than are centralized structures but they harbor the risk that malware can propagate quickly across different networks. Devices could be restricted to communicating within a geographical range on a protocol level, but this would diminish the network's resilience. Furthermore, Targon [172] argues that mesh networks are less costly to implement than standard centralized topologies, only under certain conditions. Because every mesh node requires its own end-point security, it can be assumed, in accordance with ENISA [49], that mesh networks are generally more expensive than are centralized topologies [44].

In summary the *benefits* of mesh topologies are: [44]

- *Resilience:* The effects of failures or attacks on specific nodes can be mitigated by using alternative communication links.
- *QoS:* The high number of available communication links reduces the probability of congestion, while limiting the propagation scope reduces latencies.

The *drawbacks* of mesh topologies are: [44]

- *Resource Control:* It is more difficult to establish situational awareness and optimize resources because of the dispersion of the collection process. Furthermore, the network topology may differ from the electrical topology, causing problems for control functions.
- *Security:* The high number of communication links within the mesh network supports malware propagation, enabling spreading to other critical infrastructures. Data is sent hop-by-hop through other nodes that may not be trustworthy.
- *QoS:* Mesh networks overcome local bottlenecks through load balancing. However, routing decisions and multi-hop routing can lead to additional overhead. Some routing protocols may influence latency.
- *Compatibility:* Legacy SCADA cannot easily be integrated via uplink into local nodes.
- *Cost:* The highest expenses occur at the ML-ICT level which requires extra security features. But security features have to be implemented across all nodes in the mesh.

3.4.4 Decentralized Topology

A fully decentralized architecture, cf. Figure 7-d and 11, leads to higher resilience and reduced data congestion thanks to alternative links. However, it is more vulnerable to ICT propagation, which may spread faster and even infect systems outside the power grid where similar hardware or software is in use. As mentioned in Section 3.4.3, Targon [172] and ENISA [49] argue that mesh networks are often more expensive in terms of capital expenditure, especially taking into account the cost of security functions for every node.

In summary, the *benefits* of decentralized topologies are: [44]

- *Resilience:* Local intelligence mitigates the effects of high level failure and is robust against local failures and attacks.
- *QoS:* Local data management minimizes latencies. Local control minimizes data exchange, and alternative paths prevent congestion.

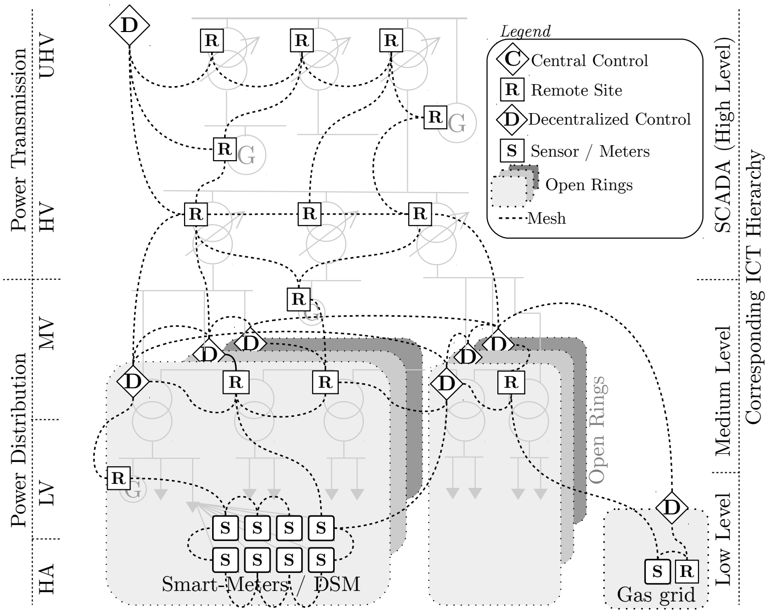


Figure 11: Smart grid with a decentralized topology, as presented in [44]

The *drawbacks* of decentralized topologies are: [44]

- *Resource Control:* Situational awareness and resource optimization are difficult to achieve because data is collected locally and must be exchanged with other nodes.
- *Security:* High connectivity between nodes and identical hard- and software facilitate the spread of malware through similar vulnerabilities.
- *Compatibility:* Extensive retrofitting becomes necessary if mesh networks are to be implemented on the higher levels.
- *Cost:* Economically inviable costs accumulate at the HL-ICT, which must be upgraded to a mesh network.

3.5 Dataset Description

The underlying data set, i.e., the parent population of data, used in this model comes directly from our industry partner [184]. It is based on the electricity grid and the respective power lines, hierarchy levels, transformer stations and loads of one central district of the city Vienna, cf. Figure 2. We consider its configuration representative for urban districts as it contains typical open rings, and dense node placement. However, we are not allowed to illustrate the data set fully or disclose information due to publication restrictions on critical node locations.

Table 8: Parent population of nodes and links, cf. Figure 12

	HL-ICT	ML-ICT	LL-ICT
Number of nodes	2	367	3078
Number of links	1[a]	62[b]	6634[c]

[a] Optical fiber connects the HL-nodes to each other.

[b] ML-HL links connect ML-nodes to HL-nodes.

[c] LL-ML links connect LL-nodes to ML-nodes.

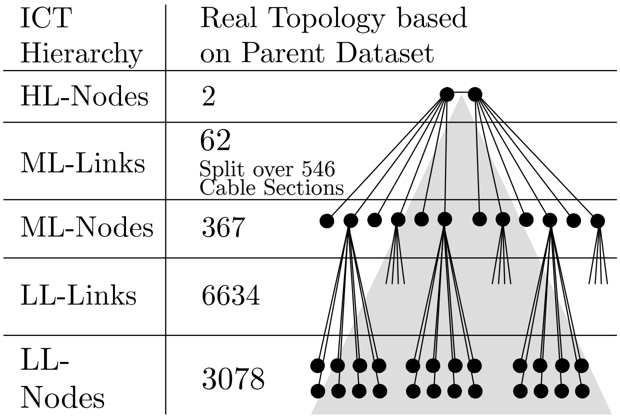


Figure 12: Derived General Topology

We extract three types of power nodes in the LV, MV and HV levels of the power grid, cf. Section 3.3. First, two high voltage transformer stations contain the central monitoring and automation equipment that is used to collect data and perform switching events. These nodes are represented as HL-nodes and share one direct communication link for updating power grid data. See Figure 12 as reference for a power grid topology illustration. Next, the data includes 367 medium voltage power-distribution transformers (represented as ML-nodes) connected to the HL-nodes via 62 medium voltage cables, thus, each MV-power line connects 6 transformers. According to [184] spare pipes and optical fiber cables are typically fitted in the power cable trenches during construction work and maintenance work. Therefore, we assume that optical communication links are either implemented or can be retrofitted with minimal effort. Since the parent data shows a number of duplicate parallel power lines for increased power transfer capacity and electrical muffs in repaired sections, we compensate those duplicates by using the geospatial information of these links and assume one continues cable per link. Finally, 3078 low voltage building connections (LL-ML links) directly connect the LL-nodes to the medium voltage transformers (ML-nodes) via 6634 cables including duplicates and muffs. We apply the same method used with the ML-nodes to compensate. Figure 14 and Tables 8 and 9 present

Table 9: Parent population extreme values and percentiles of link lengths

Distance in [m]	LL uplink	ML uplink	HL-links
Maximum value	346.1	2662.9	812
95th percentile	132.0	884.2	n.a.
75th percentile	67.2	351.2	n.a.
Median	38.3	201.7	n.a.
25th percentile	19.8	124.8	n.a.
5th percentile	6	52	n.a.
Minimum value	0.3	23.9	n.a.
Average	49.4	304.2	n.a.

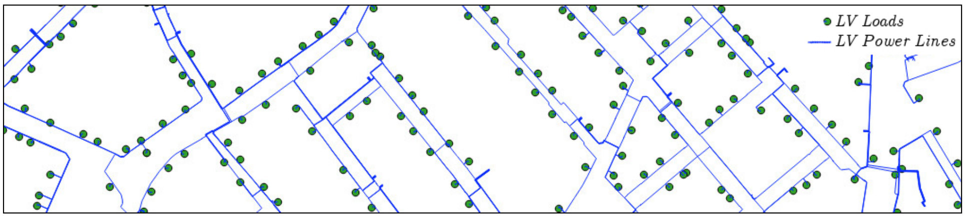


Figure 13: Geo Located Data Excerpt; Random Sample

details on the parent data set.

Figure 13 illustrates a small random section of the original data set, including LV building connection nodes and their corresponding power lines. The remainder, includes critical nodes and power lines which are redacted as mentioned above. Therefore, we abstract the critical geospatial data in Figure 14 while retaining the relevant distance information of all link lengths.

3.5.1 Real Topology based on Parent Data Set

The parent data shows that all nodes and links are distributed approximately equal between the two HL-nodes. Therefore, we assume a symmetrical general topology, for further modeling. Figure 12 illustrates the derived real topology from the extracted data. We generally assume that all links between HL-nodes and ML-nodes are realized with Wavelength Division Multiple Access (WDMA) based optical fiber. Therefore, all ML-nodes are connected to their respective HL-node via a dedicated channel inside the shared optical fiber. The reason for using WDMA is elaborated in Section 5.2 for all scenarios utilizing wired link technology. Furthermore, we assume PLC over Frequency Division Multiple Access (FDMA) in the low levels of the topology, because we find it unfeasible to connect every household with a dedicated optical communications link in the near future. This allows us to model the LV-links as a centralized topology for those scenarios with wired technology. Section 5.2 elaborates on PLC.

3.5.2 Dwellings per Building

According to Statistik Austria [167] the number of dwellings allocated to the sample of the city district in question, amounts to 18806 dwellings per 1583 buildings. Therefore, the average amount of dwellings per building is 11.97. We conservatively assume 12 dwellings per building, thus, each LL-node represents one building manager node, transmitting with the communications pattern of 12 collective smart meters. Additionally we add one decentralized renewable energy manager to each LL-node to represent the smart grid capabilities. The transmission patterns are discussed in detail in Section 3.3.

3.6 Statistical Abstraction

In this section we derive statistical data from the original data set as summarized in Table 8. The average wired link distance is calculated from the cable lengths in the parent data. We compensate for any aforementioned power line duplicates by ascertaining that they are spread evenly throughout all hierarchy levels and therefore have no effect on the statistic. The air-distance between nodes however, is calculated with a distance matrix, generated from node clusters in the parent data set. We choose the upper and lower limit for both wired and wireless statistics at 95% and 5% respectively. We consider the rest as outliers, shown in Figure 14.

3.6.1 Wired Communication

The wired link distances are based on the geospatial power line data discussed in Section 3.5. According to Table 8 the ML-ICT consists of 367 nodes and as many links. As mentioned above, we assume optical fiber links that are fitted parallel to the MV power lines. Furthermore, we assume that every ML-node will be fitted with a standard communications unit, including amplifiers for PLC and optical fiber technology.

Table 9 elaborates on the extreme values of this dataset. We limit the dataset to the upper and lower 5% of all values. Figure 14(a) shows that 95% of all distance values among the medium level ICT nodes are smaller than 885 meters, which becomes the upper limit. The average wired distance is 304.4 meters. The low level ICT node distances are arranged more densely at 95% of all values smaller than 102 meters. The average distance is 38.2 meters.

Table 10: Wired communication, extreme values and percentile link lengths, derived from Table 9

in [m]	LL uplink	ML uplink
Maximum value	346.1	2662.9
95th percentile	132.0	884.2
75th percentile	67.2	351.2
Median	38.8	201.7
25th percentile	19.8	124.8
5th percentile	6.0	52.0
Minimum value	1.0	23.9
Average	49.4	304.2

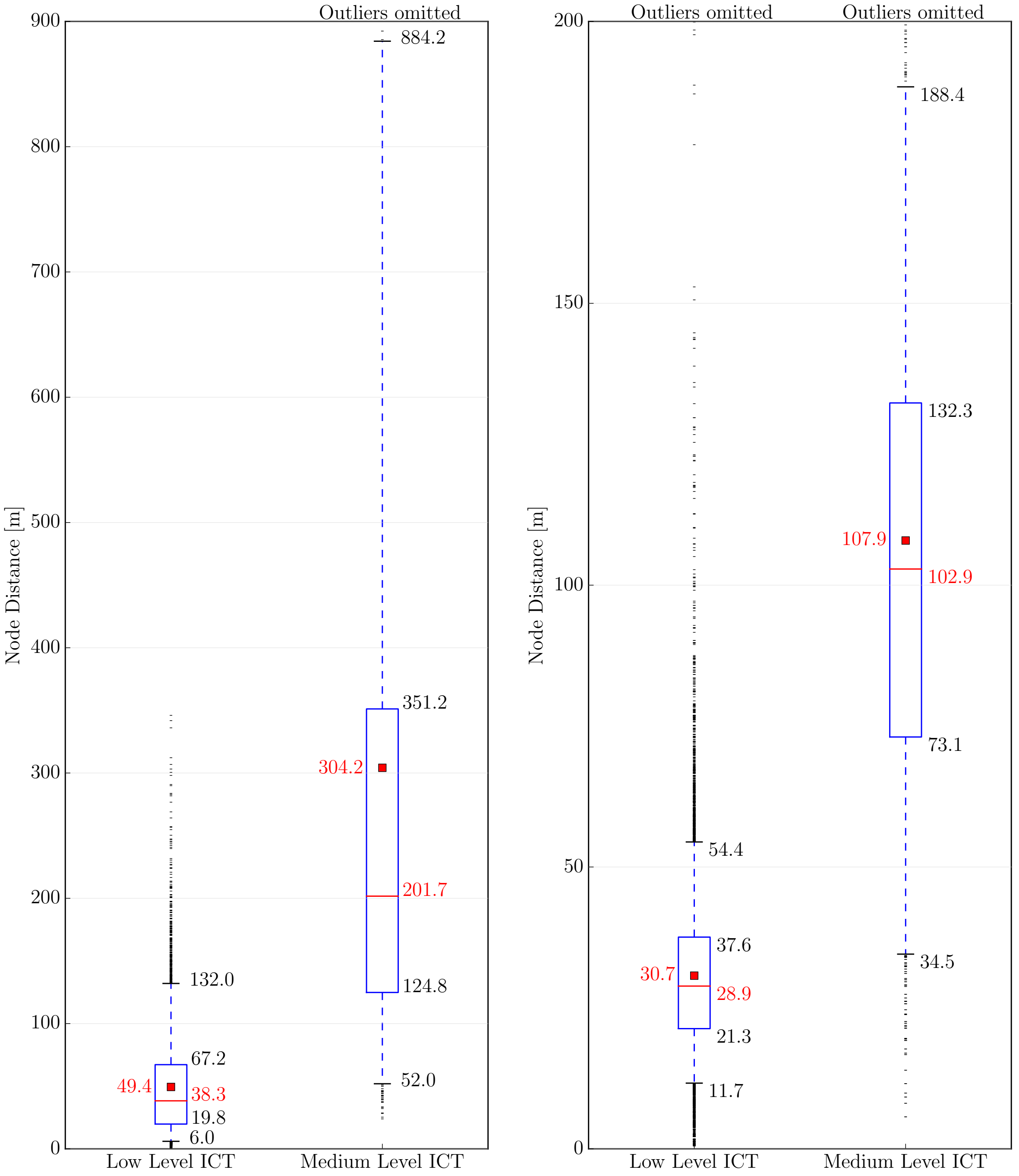
3.6.2 Wireless Communication

We consider wireless mesh technologies in some scenarios, and base it on the Open Link State Routing (OLSR) protocol. Since we are not allowed to publish the node-locations in the parent data, we resort to obfuscating the data by means of generating a distance matrix from the existing nodes in Section 3.5. We classify the nodes into groups and analyze the average distance between the 5 nearest neighbors of each node, because, according to the OLSR-Manpage [139] there is a maximum of 5 simultaneous connections per node. We use the software QGIS [140] to generate a distance matrix. This method results in a list of over the air distances for the wireless mesh communication. Figure 14(b) illustrates both the LL-ICT and ML-ICT data.

The generated matrices contain 1895 distance values for the ML-nodes and 15390 for the LL-nodes. Even the most extreme outliers at 437.7 meters, lies within the physical maximum node distance of 500 meters, according to [91, 92]. Table 11 elaborates on all values of the distance matrix dataset and Figure 14(b) illustrates them. The mean value for ML-ICT is 102.9 meter.

Table 11: Wireless communication, extreme values and percentile link lengths

in [m]	LL uplink	ML uplink
Maximum value	268.7	437.8
95th percentile	54.4	188.4
75th percentile	37.6	132.3
Median	28.9	102.9
25th percentile	21.3	73.1
5th percentile	11.7	34.5
Minimum value	0.5	5.7
Average	30.7	107.9



(a) Wired link distances from the parent population (b) Wireless link distances as per the distance matrix

Figure 14: Statistical abstraction

4 Malware Model

Notice of adoption from previous publications:
Parts of this section have been previously published in [48], which introduces a generic malware-based attack life-cycle, several different malware-types, and attack capabilities for smart grid environments, among others. Furthermore, three superclasses of malware derived from existing technologies are expanding our scope. The authors main contribution was the development of the malware superclasses, including an extensive investigation of different malware types, and the generic life-cycle model. The co-authors contributed valuable input on the evaluation metrics in discussions and challenged their applicability. The attack capabilities and malware life-cycle were also presented in [38, 40, 45].

This chapter represents our theoretical malware model. It includes the attack life-cycle, divided into the major stages malware has to pass through, detailed information for every stage, and our malware superclasses.

4.1 Methodology

We conduct an extensive investigation of 19 existing malware types and use them to extract all features relevant for attacking smart grid control networks. We use the results from our generic attack life-cycle to implement our malware superclasses which represent the most important existing malware types. We cover research question 2, cf. Section 1.3, with this analysis. Additionally, we extend our focus by a list of defense measures, published in [41, 43]. Should real implementations of our malware superclasses be deployed in the real world, significant system failure in existing control networks could be the result. Therefore, real experiments in operational control networks are not an option for this work.

4.2 Smart Grid Attack Model

Many of the malware types we investigated, cf. Section 8.1, are APT supported. Therefore, our attack model is based on well financed and highly skilled adversaries. However, we also take less-equipped attackers into account, who are able to reorganize publicly released malware and retrofit new features. Our attack life-cycle model, therefore, intentionally includes many conceivable attack vectors, familiar from the Internet landscape. However, we exclude malicious insiders, because they can best be defended against by increased awareness and strict company policies, e.g., user management, rather than technical means. We assume the attackers are capable of utilizing zero-days and operate on distributed network resources. We present several malware types and their properties in Section 8.1, as well as discussing their impact on smart grids. Although some do not utilize zero-days, we assume incorporating their features in more capable malware is feasible and will present a major challenge in the future. Furthermore, we assume that attackers cannot interfere with properly implemented defense mechanisms, except for known evasion techniques that are discussed in Section 8.1.4 [48].

4.3 Smart Grid Security Measures Baseline

Well-established security guidelines to prevent predictable attacks exist from classical Internet security, as do standards on power control equipment, and smart grid implementations. Although the reality in existing installations is still alarmingly deficient, for our analysis we assume that these measures are in place. The German Federal Office for Information Security (BSI) [61, 62] has developed a network architecture based on a smart meter gateway that significantly narrows the attack surface. However, these strict rules for smart metering do not protect other types of devices and services hosted in smart grids. PMUs or other sensors often operate time-critical services that cannot tolerate retransmission. With the use of network segmentation for instance, a clear separation of the objective of such services is possible. Furthermore, field devices have to cope with an increasing number of vulnerabilities over time [189]. Therefore, we expect that attacks on power infrastructures will become more common upon the implementation of automated smart grid control in the coming years. We aggregate several guidelines from [8, 61, 62, 128, 174, 189] and summarize the most important of them. In some countries, these are compulsory by law, in others they are only suggestions. We also discuss additional security measures in Section 10 for future examples [48].

- *Security updates / update policy:* Regular and timely updates for devices in business and industrial networks prevent vulnerabilities and minimize the window of opportunity for attackers.
- *User management*: All users, including administrators, are restricted to environments and have capabilities available, suitable for their role.
- *Password policy*: Strong password policy ensures the use of long non-repeating high-entropy pass-phrases.
- *Anti-virus*: Modern anti-virus software, i.e., anti-malware tools, are used, that should be based on heuristics and remote reputation services.
- *Network segmentation*: Subnetworks with distinct objectives are sep-

arated, e.g., segmentation exists between administrative network and industrial control environment.

- *Restricting remote access*: Since segmentation can be circumvented by remote access, the latter is strictly controlled and limited to trusted parties, if permitted at all.
- *Strict firewall rules:* All access is prohibited by default except for whitelisted hosts and services, protecting users from Internet threats.
- *Decentralization of critical services:* Decentralization strengthens resilience against failure and attacks. However, explicit countermeasures are required to counteract propagation methods that do not depend on a functioning network, e.g., infected removable drives.
- *Dimensioning hardware for future software updates*: Smart grid devices remain in service for more than 10 years. Whenever resource-constrained hard- or software is integrated into modern equipment as part of a modular design, the entire system security may be compromised with regard to sophisticated attacks. Therefore, these devices are prepared for future demands and provide sufficient resources to support updates.
- *User education*: One of the most basic preventative measures is user education, which protects hosts and their users against many simple access vectors. In combination with strong passwords, this can significantly impede propagation.
- *Ensuring data confidentiality, integrity, and availability (CIA)*: The correct implementation of standard protocols prevents, e.g., packet-integrity-attacks or sniffing.
- *Business continuity plans*: A tailored process that allows for partial operation of the system in fall-back mode, i.e., reduced services or emergency operation, must include organizational and technical measures that support the recovery process.
- *Hardening of operational assets*: The operational assets, e.g., servers, firewalls, and switches, must be hardened against well known attacks that can be expected in future.

4.4 Malware Abstraction: A Generic Attack Life-Cycle

This section proposes a generic model for multiple stages in the life-cycle of cyber-attacks and malware communications. We consider the existing approaches discussed in Section 4.3 and reuse some of their concepts in our

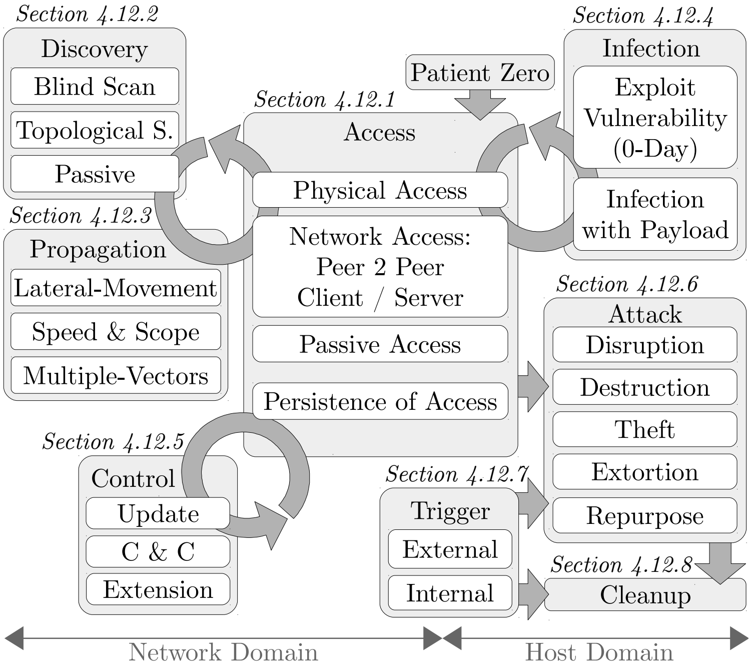


Figure 15: Generic stages of malware-based cyber-attacks, as presented in [48]

model. However, rather than confining our model to a strictly linear approach, we argue that loop-back cycles more accurately reflect the greater flexibility that is intrinsic to modern malware. All phases in our model revolve around *access* to resources. The proposed model begins with a *discovery-propagation-access* cycle for the network side of target-discovery and propagation. This is followed by an *infection-access* cycle for host infection and privilege escalation. A *control-access* cycle represents the remote-control infrastructure that allows for functional updates, as in C&C-triggered updates and remote commands. Subsequently, the model proposes *attack*, *trigger*, and *cleanup* stages, which are deployed once sufficient access to the critical resources is attained and the attacker is capable of commencing the desired attack. Figure 15 and the following sections provide detail on all stages:

- *Access*: The centerpiece of any attack is direct or remote access to critical resources. Initial access is often of inferior quality and requires privilege escalation for further attack commencement. When administrator access to a resource is available, further propagation or an attack may be possible. Applicable access methods are summarized in Section 4.4.1.
- *Discovery*: If insufficient access is available to achieve a specific goal, discovery and scanning methods are used to locate new victims in the

network. Attackers, thus, gain additional knowledge in the mapping process. Techniques range from noisy to covert scanning and are summarized in Section 4.4.2. Topics of interest include security by design and heterogeneity of devices.

- *Propagation*: After discovering new targets, malware propagates to new hosts using exploits. Once transmission is complete, the malware has gained access to the target. Propagation techniques range from direct connections, to highly covert ones, as elaborated in Section 4.4.3. Details concerning transmission can also be found there. It is worth noting that all propagation techniques are also suitable as access techniques.
- *Infection*: After gaining access to a new host, malware can infect and escalate initial user rights to a higher level, increasing its access quality. Section 4.4.4 contains details on this.
- *Control*: Most modern malware implementations are controlled externally, via C&C instructions. In addition to loading new modules and controlling the spread as seen in many cases [56,96,97,123,169], other methods are described in Section 4.4.5.
- *Attack*: Successful attacks depend on sufficient access to compromised critical resources in the network. The direct transition from the access block to the attack block in our model reflects this dependency. A variety of types of attacks are possible including service disruption, physical destruction, data theft, espionage, extortion, or repurposing (cf. Section 4.4.6).
- *Trigger*: Complex attacks require coordinated action and orchestration. Attack triggers can be hard-coded or remotely activated, as described in Section 4.4.7.
- *Cleanup*: Many adversaries, especially on APT-level, conduct covert operations [95,98] and may be interested in concealing their technology. Therefore, hiding their tracks by removing or encrypting parts of the malicious code, e.g., modules, can be effective in combination with other persistence mechanisms as discussed in Section 4.4.8.

4.4.1 Access

In this section we provide detail on access methods to resources such as hosts and networks. Furthermore, we point out similarities among those access methods that are also used as propagation methods. Sufficient access is a prerequisite for all other blocks in our generic attack-model. At the end of this section, we discuss the possibility of long-term persistence of malware in spite of the implemented countermeasures. Initially a foothold inside the network is established, often via an office PC, i.e., patient zero, in the enterprise network. After infection, the attacker escalates its privileges to a higher level of access by exploiting a vulnerability allowing for additional steps. These include *discovery*, discussed in Section 4.4.2, and *propagation*, discussed in Section 4.4.3. We generally distinguish between host and network access. However, both are required to successfully infiltrate networks.

This list summarizes the considered access vectors and provides examples:

- *Physical access* refers to methods by which an attacker reaches the target host directly and modifies hard- or software. Examples include direct data manipulation via Universal Serial Bus (USB) drives, hard drive exchange, live disc reboot or direct installation of malware.
- Active *Peer-to-Peer (P2P)* network access refers to P2P protocols and file-sharing, in which each host acts as a server and a client at the same time.
- Active *Client / Server* network access designates typical structures with separated clients and servers.
  - *Server-to-Server (S2S)* access refers to the use of vulnerabilities that enable the infection of other servers, for instance via buffer overflow attacks to insert backdoors or server exploits [131].
  - *Server-to-Client (S2C)* access includes all methods that allow compromising a client system from an infected server, e.g., watering hole attacks [155] where groups of users are targeted by compromised web services or rapid client reinfection by persistently infected servers [96].
  - *Client-to-Server (C2S)* access refers to all methods which, for instance inject exploit code into websites [23].
  - *Client-to-Client (C2C)* access allows the infection of other clients via remote code execution or auto-run shared files, without infecting the server [94].
- *Passive access* includes methods by which the attacker gains remote control over resources through social engineering and deceptive abuse of a persons trust:
  - *Host-to-Network-Share (H2NS)*: This group subsumes all access vectors that utilize vulnerabilities to infect files in trusted network shares, including backup drives and shared folders [57].
  - *Removable Drives (RD)*: Infected USB drives, external hard drives, or other removable media provide effective propagation and access methods by utilizing auto-run exploits on the target

host. They can, for instance, be strategically placed so they attract individuals who unknowingly infect the target network. Another example is the infection of employee USB drives by the host. These methods benefit from being invisible on the network, yet exhibiting effective reach across air-gaps. Some sophisticated malware even disinfect removable drives in order to increase their stealthiness [98].

- *Phishing emails*: The bulk dispatch of emails containing infected attachments or malicious web links is a method heavily used to gain initial access. Some malware uses phishing alone for propagation, but most resort to alternative methods after establishing a foothold. Several email protocols are used in phishing methods and victims typically have to open the attachment or web link to commence the infection process [89].

Faulhaber et al. [59] argue that passive access vectors, which require user interaction, rank among the most effective. They account for about 88% of all Microsoft Windows-based system infections. Of these, phishing emails with malicious attachments or web links account for approximately 45%, auto-run features on removable drives for another 26% and H2NS infections for 17%. The authors conclude that exploits on non-updated hosts account for less than 6% and zero-days even lower.

Although zero-days seem to be rare, it is a mistake to infer from this that the need for countermeasures is in any way reduced. Advanced malware that utilize them have been observed in the wild on several occasions [13,56,97,98] and examples are expected to be seen more frequently in the future. Since substantial resources are required for their development, adversaries capable of funding such campaigns can afford and often do utilize zero-days. They can immediately unhinge access rights restrictions, granting the attacker system access at will.

Lee et al. [113] elaborate on APT-made malware based on an analysis of a recent cyber-attack on the Ukrainian power grid, that occurred in December 2015. They found that phishing emails were used for initial access, then a backdoor established C&C which allowed further propagation inside the network. After establishing persistence, stealing certificates, and creating administrator accounts the attackers pivoted to the control network of the power grid. Section 4.4.6 elaborates on the extensive attack methodology, confirming that standard security measures do not suffice to repel such advanced attackers [57, 94, 97, 98].

Physical access to field devices is one of the most obvious means of entry specific to smart grid control networks. Furthermore, two-way communication between field devices and servers opens vectors toward higher levels in the

ICT hierarchy. These field devices are connected to a critical network that must be protected using adequate Intrusion Detection Systems (IDS), which can identify and mitigate malware spreading. Border gateway protection is insufficient once attackers are inside the network and can move freely. This is the case if physical access or secondary targets such as trusted partners including external services provide lateral access vectors, as seen during the Aurora attack [109]. Therefore, intrusion and anomaly detection across strictly segmented sub-networks becomes increasingly important [44, 113, 174].

Concerning *Access Persistence* we discuss defensive methods malware authors use to ensure continuous access, even beyond security measures implemented on the host. Based on [13,31,57,58,72,89,94–96,98,99,123,169,175], many malware types establish persistence using a variety of methods. These include, but are not limited to, stealing credentials or injecting malicious code into host core processes, e.g., local drivers. Beyond that, malware often uses anti-detection mechanisms such as code obfuscation, encryption, memory residency, or detection of anti-virus software, henceforth referred to as anti-malware tools. The capabilities of modern malware are increasing in complexity, transforming them into multipurpose attack platforms. This development is true for both host-based and network-based access forms including C&C, scanning, remote code execution, and propagation.

Based on the aforementioned sources, we can list several persistence methods:

- Manipulation, or deactivation of host-based anti-malware tools helps the malware evade detection.
- Code obfuscation increases stealthiness.
- Multi-layer encryption techniques increase stealthiness.
- Local recompilation changes the appearance of malware.
- Rapid reinfection upon disinfection increases persistence.
- Credential theft and exploit allows administrator access.
- Memory residency helps in evading detection.
- Service injection into system processes can survive restart.
- Cleanup mechanisms prevent forensic analysis.

4.4.2 Discovery

This section discusses known discovery techniques such as network scanning, as well as the protocols in use. Sufficient access to a host and local privileges are a prerequisite for discovery. Consequently, scanning for new victims precedes the propagation block in Figure 15. Li et al. [114], Staniford et al. [166], and Riley et al. [147] discuss several scanning methods and cluster them into the following categories:

- *Blind scanning* targets randomly generated address ranges and commences sequential, clustered or random scanning. This approach produces a high rate of failed connections, thus, anomalies, which are easily detected. Malware that utilizes blind scanning generally spreads fast, yet imprecisely [114,147], as seen in our simulation results, cf. Sections 9.3.1 through 9.3.4.
- *Routing scan* methods use the Border Gateway Protocol (BGP) to decrease the scanning space, resulting in a better hit rate and lower background noise, which can be used by anomaly detection algorithms. Routing scans generally target countries or regions [114, 147].
- *Topological scanning* improves the hit rate further by obtaining information about the target network from the host. It operates more stealthily and produces even fewer anomalies. Examples mentioned in [166] show that CodeRed2 is capable of preferring local networks even in a semi-random-scan. Furthermore, it scans outside its local address space only with certain restrictions, producing fewer anomalies than its sibling CodeRed1 [114, 147].
- *Passive scanning* does not probe the network actively but rather waits for native connections to be initiated. Such malware types spread slowly, however, no scanning anomalies are produced. Infectious packets may be sent by initiating a session with known hosts, yet could trigger anomaly detection. Alternatively, covert attachments onto active transmissions may decrease the anomaly output further. We simulate such a case and present the results in the Contagion parts of Sections 9.3.1 through 9.3.4. Such a threat may spread faster in a homogeneous environment, e.g., smart grids, or P2P-networks, where a host is likely running a single dominant implementation [114, 147, 166].
- *Hitlist scanning* requires a list of initial targets as a prerequisite. This list may be created and updated through the use of a botnet or by collecting additional information from the infected hosts. Hitlist scanning produces few anomalies in a network increasing stealthiness. It is a technique for accelerating the initial spread, which is an important measure for the early stages of infection. We simulate such a case and present the results in the Endemic parts of Sections 9.3.1 through 9.3.4.
- *Permutation scanning* is an augmentation for histlist scanning that distributes the target list between parent and child malware. This method increases stealthiness by decreasing the amount of rescanning, each in-

stance appearing to scan randomly and separately. Hitlist and Permutation Scanning make a malware fast enough to attack most vulnerable hosts Internet-wide in under one hour [114, 147, 166].

- *Distributed scanning* was described by Dainotti et al. [31] and Staniford et al. [166]. An example of a large-scale stealth scan showed that approximately 3 million infected hosts were used to scan the Internet Protocol Version 4 (IPv4) address range randomly in reversed byte order. This method was considered impressively stealthy, as hardly any source addresses rescanned a similar address range. Therefore, few anomalies are visible in traffic. Compared to older scanning methods which generally originate from a small set of sources, this method is more capable of scanning covertly thanks to its sheer size.

Typical techniques used for scanning in the IPv4 address space utilize the following protocols: User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Address Resolution Protocol (ARP), File Transfer Protocol (FTP), or Internet Control Message Protocol (ICMP). See Roger and Tan [149, 171] for details on their scanning methods: Vanilla TCP Connect Scan, TCP SYN (Half Open) Scan, TCP FIN Scan, TCP Reverse Ident Scan, TCP XMAS Scan, TCP NULL Scan, TCP ACK Scan, UDP ICMP Port Scan, ARP Scan, FTP Bounce Scan, and ICMP Ping-Sweeping Scan.

With Internet Protocol Version 6 (IPv6), active scanning is futile due to the large address space. According to RFC5157 [24], even search space reduction techniques can prove futile, although passive scanning could be used instead. An attacker could do so by hosting a harmless and free web service, to attract unsuspecting users [155] in a watering hole attack. IPv6 addresses of target hosts, including ones that are generated with the privacy extensions, can then be extracted from the generated logs. This method was used by the security search engine Shodan [138, 159] by contributing time servers to the debian Network Time Protocol (NTP) pool [179], and was discovered in 2016. Every network packet to one of the contributed servers resulted in a network scan of the origin. Even the limited lifetime of IPv6 privacy extensions (generally one day) provides a large enough window of opportunity to infect a host.

4.4.3 Propagation

This section elaborates on propagation methods between hosts via non-network or network transmission. Knowledge of potential victims is required and generated during the discovery phase, cf. Section 4.4.2, after which successful propagation to a new host allows low level access, cf. Section 4.4.1. Propagation then leads to wider/higher quality access and ultimately better knowledge of the environment, which is essential for commencing attacks.

All methods, except for physical access as discussed in Section 4.4.1, are usable as propagation methods. The decision *not to* utilize a method is an implementation choice of the malware in question, which can refrain from using a technically feasible option. For instance, Locky [162] utilizes email for initial access but not for propagation. Analog to Section 4.4.1, we generally distinguish between active and passive propagation:

- *Active* methods enable self-propagation, usually by exploiting remote code execution or auto-run features. Active propagation is generally very fast and requires no user-interaction. They include P2P, S2S, S2C, C2S, and C2C.
- *Passive* propagation, on the other hand, can be slower yet is more effective at bridging air-gapped networks, e.g., ones in which users physically carry removable drives into segmented networks and infect hosts. Many malware types nowadays have air-gap capabilities [58,95–98,123], effectively moving and communicating between seemingly secure networks. Propagation via H2NS or phishing emails requires user interaction in the form of opening files, attachments, or web links.

With regard to secure networks, [13, 59, 166] elaborate on cyber-attacks increasingly targeting homogeneous and widely used services. Propagation can commence across vast distances in rapid succession, as displayed in aggressive malware types such as Slammer [114] or CodeRed2 [23]. Man-in-the-middle attacks on monocultures are also familiar from Flame [97], whose fake update servers infect hosts (S2C). Aside from anomaly detection, no feasible network based method exists for ensuring the intended behavior of hosts. There are, however, a number of host-based detection mechanisms that are resource-intensive, for which field devices should provide adequate resources. One particular application for smart grids concerns the smart meter update services that should also work on older models, i.e., the hardware should support future updates. According to Li et al. [114], malware generally consists of two parts, a payload and a dropper, that may propagate in one of the following transfer schemes:

- *Self-carried*: Payload and dropper are transferred in one file-set. Typical examples are Code Red 1 and 2 [23,131,147], and Nimda [23,166].

- *Second-channel*: The payload is downloaded later through a backdoor, following an infector, i.e., the dropper file. Typical examples of second-channel malware include Regin [169], Duqu, Duqu2 [13, 96, 173], and BlackEnergy3 [56].
- *Embedded*: The payloads differ for each instance and masquerade communication as normal traffic, as mentioned by Staniford et al. [166].

According to [96, 98, 109], modern malware increasingly utilizes *modular* propagation schemes such as second-channels that only acquire the modules needed at any given time. They also allow multiple propagation paths, which makes them more diverse, cf. Table 27. Only a few utilize either passive propagation or an unknown embedded method that may depend on the use of *covert channels*. This, however, renders them invisible to network-based detection. Staniford et al. introduced such a method in their "contagion" model [166]. Real world examples include Locky [9,162], Gauss [13,98], and Equation [95] which are known for their prolonged covert behavior. There are indications that Gauss utilizes zero-days for propagation, which may imply the existence of unknown covert techniques. [48,95]

All malware propagation relies on common network- and transport-layer protocols such as Internet Protocol (IP), UDP or TCP. The choice of protocol can have substantial influence on the propagation characteristics and performance:

- *UDP*-based malware communication can saturate the maximum link-bandwidth [114, 147]. Such communication often has a small packet size, which enables fast infection. UDP is connectionless, meaning that packets may be dropped or denied leaving no guarantee of delivery. However, broadcast in the network commences regardless. Examples of malware capable of UDP transmission are Nimda [23,166], Slammer [114,147], Sality [31,175], Conficker [31,59,158], or Regin [169]. Except for Slammer, the others also use TCP, employing each protocol for different parts of the scanning and propagation cycle [31].
- *TCP*-based malware has the advantage of reliable transmission but is constrained by the number of parallel connections and certain latency limitations. Through parallelization the number of connections may be increased, and simultaneous infection becomes possible. Since TCP connections require a handshake, these malware types are limited by the average Round-Trip Time (RTT) of a network. Real world examples include: Code Red 1 and 2 [23, 131, 147], Nimda [23, 166], Sality [31, 175], Conficker [31, 59, 158], Regin [169], Aurora [95, 123], Stuxnet [58, 123], Duqu and Duqu2 [13, 96, 173], Flame [13, 97], BlackEnergy3 [56], CozyDuke [57], and PLC-Blaster [108, 165].

4.4.4 Infection & Exploit

According to Li et al. [114], most older malware types only target a single operating system and few vulnerabilities. However, Matrosov et al. [123], Symantec [169], and Bencsath et al. [13] show that in the recent past, digital warfare has evolved into a more complex environment. Attacks involving multiple exploits, infection vectors, and payloads are common and must be considered to be the state of the art. NIST [133] and Mitre Corp. [129] provide lists of known vulnerabilities containing a number of possibilities to exploit all kinds of operating systems.

Increasing modularity in modern malware also allows for a multitude of payloads, thus adding flexibility; for instance, Duqu has only six modules whereas Duqu2 has over one hundred modules [13, 96], increasing its range of possible uses.

According to a number of sources [56,57,95–99,123,169], most APT-authored malware utilize zero-days. They are, due to their undisclosed nature, nearly impossible to defend against, which makes them a valuable asset for attackers. This demand has lead to black markets for zero-days. The defense systems for critical infrastructures must, therefore, be designed to repel skilled attackers, or at the very least segment in such a manner that attacks can be mitigated. Well-understood threats may be circumvented by standard methods, but deterring APTs requires elaborate security considerations.

Ször et al. [170] and Li et al. [114] categorize malware according to three distinct payload types. Additional features are added to malware according to the following methods:

- *Monomorphic* malware may vary in size through the use of padding and evade detection by fragmentation. Yet all instances produce the same signature and are easily detectable using anti-malware tools. Examples include CodeRed1 and 2, Nimda, Slammer, and PLC-Blaster [23, 114, 131, 147, 165, 166].
- *Polymorphic* malware types scramble the payload through encryption. Therefore, every instance has a different signature and size. However, when the payload or parts of it are decrypted on the local host, every instance has the same signature. This method provides better evasion properties, yet may be detectable using sophisticated anti-malware tools. Examples include Conficker, Regin, Stuxnet, Duqu, and BlackEnergy [13, 31, 56, 58, 59, 94, 123, 158, 169, 173].
- *Metamorphic* malware are able to create new instances that appear to be different from their parent. They vary in shape, size, encoding, and encryption and utilize recompilation on the host system to change

their appearance. They do not carry a decryptor; instead, all data is carried in one single code body. These types of malware are notoriously difficult to detect. An example is AdWind [89].

4.4.5 Control

Modern malware is generally controlled by a C&C infrastructure, which allows the botmaster to influence them and also enables modular extension and updates. Although, IDS try to match C&C traffic patterns found in the communication, signature-based systems such as Snort [148] can be defeated by encrypting communication. Therefore, malware such as BlackEnergy, that used clear text for C&C traffic in early versions [135], employs full encryption in the latest version [56]. Anomaly detection systems may, however, still be capable of detecting suspicious encrypted traffic.

The outcome of our literature review suggests that non-encrypted or non-obfuscated C&C methods have largely been replaced by fully encrypted and obfuscated methods [13, 56, 57, 169]. Some examples even utilize complex multi-layer encryption [72] or highly obfuscated methods [96] which can now be considered to be the state of the art. Another method that helps obfuscate C&C is the use of highly distributed C&C architectures to avoid detection, as do Regin, Aurora, Equation, and Locky [95,162,169,171].

Recent developments, however, push the boundary toward covert communication. Mazurczyk et al. [126] and Kaspersky [98] discuss such behavior in modern malware. While Duqu can attach itself to harmless communication using JPEG images, Regin recently also acquired covert capability. It seems that covert channels are now gaining momentum toward becoming economically viable as most outbound traffic is allowed to pass unrestricted: For instance, ICMP traffic can be abused by entering information into the data fields of echo requests and replies. However, today many routers block ICMP traffic for security reasons. While ICMPv4 can be blocked completely, critical functions in ICMPv6 prevent routers from blocking it. RFC4890 [33] and RFC2979 [65] discuss firewall guidelines for IPv6 and IPv4. Even though covert techniques are not yet very sophisticated or wide spread, development is commencing, making them a future threat.

Several sources [9,13,23,31,56–58,72,89,94–98,108,123,131,147,158,162,165,166,175] elaborate on what protocols, interfaces, and evasion techniques modern malware utilize for C&C. These protocols include: Internet Relay Chat (IRC), ICMP, Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol Secure (HTTPS), P2P protocols, Virtual Private Network (VPN), and Server Message Block (SMB). The following have been used as interfaces for C&C architectures: WinAPI, USB, Network Pipes, Mailslots, and backdoors of older malware. Several methods, including, for instance, Domain Name System (DNS) flux, IP flux, encryption, obfuscation, highly distributed C&C architectures, and covert channels have been used as evasion techniques.

As mentioned above, modern malware may use VPNs stacked on top of other protocols. Regin [169], for instance, supports custom UDP, TCP, SMB, network pipes, HTTP, HTTPS, and ICMP protocols as its base. The protocols are negotiated between infected nodes whereas the control messages use the overlaid VPN. For this reason the observable malware traffic disappears within native traffic encountered in the network. Regin is not only modular, but written from scratch as a service-oriented architecture [87]. This means that modules on the same host have distinct VPN addresses that can only be controlled via network communication. This potentially allows the attack to be distributed over multiple hosts, where every host just carries out a small part, reducing the visibility.

4.4.6 Attack Methods

This section discusses and categorizes several attack methods. We consider all levels of complexity, however, focus on APT-orchestrated attacks, as they are hardest to defend against. Li et al. [114] point out that high-complexity attacks are expensive to develop and difficult to coordinate. APTs, however, are capable of investing substantial funds in the development of the required methods. As a consequence, defending against them becomes increasingly costly and difficult. Furthermore, the defender's response time may be slow due to the increased complexity in comparison to less sophisticated attacks. Multiple vulnerabilities lead to greater damage in a shorter time and even though such attacks are of a low probability, they are, according to Line et al. [115], not to be underestimated.

According to Bencsáth et al. [13], an adversary has the advantage of using common-off-the-shelf products and choosing from a number of methods to fine-tune an attack. Furthermore, cyber-attacks are usually conducted anonymously, and, in the case of failure, there are few consequences for the attacker. However, defenders have to fend off all possible attacks, which leads to asymmetries of knowledge. Defenders can, in the best case, mitigate an attack, but never win against anonymous attackers.

We distinguish five general attack goals against networks and communication technology in this section. We will elaborate on them, including examples of smart grid attacks in Section 8.3.

- *Disruption attacks* aim to suppress a service or the production of a commodity for the period the attack lasts, e.g., flooding a target with unsolicited queries. Such DDoS attacks usually cause no physical damage, yet can result in monetary loss or outages. Real-world examples are discussed in [23,114,131,147,166].

- *Destruction attacks* generally target industrial equipment, with the goal of destroying its infrastructure, effectively halting production for a prolonged period and necessitating replacement. We examined two well-known examples, namely the Stuxnet and Ukraine attacks [56, 123]. The authors behind Stuxnet targeted uranium enrichment centrifuges mechanically destroying them by manipulating revolution speeds. BlackEnergy3 managed to destroy Ukrainian power management equipment, leaving parts of the power grid without automatic-restore functionality. This attack was mounted in multiple stages. Legacy communication equipment and industrial computers were manipulated on a firmware-level simultaneously, to prevent automatic recovery and restoration processes. The backup batteries were discharged and the hard drives deleted, thus, servers failed to function. Finally, essential power switches on the transformers were opened remotely, and a DDoS attack was mounted against the telephone hotlines preventing customers from reporting the power blackouts. Furthermore, this incident provides a blueprint for imitation.
- *Theft* likely ranks among the most common attack types according to descriptions in [95, 96, 98, 158, 169, 175]. These espionage campaigns range across all sectors, from companies and industries to politics, academia, or military. Data theft is typically conducted over long periods and with very low visibility.
- *Extortion schemes* have recently made a resurgence in the form of crypto-lockers. During such attacks host-devices are infected, all accessible files encrypted and all backups removed. Generally, local and remote files are affected, leading to a broad impact through shared network folders. Ransom is demanded from the victim to obtain the decryption key, as seen in the case of Locky [9, 162]. There are, however, several other possibilities for extortion of operators of critical infrastructures, as discussed in Section 8.3.
- *Repurpose attacks* change the behavior of a host from its intended function; for instance, infected hosts act as stepping stones or proxies to obfuscate the attacker's infection and C&C paths.

4.4.7 Trigger

Triggers for executing attacks may be hard-coded (internal) or remote-controlled (external). One particularly interesting example of a hard-coded trigger is the "Gödel" module in Gauss [13, 98], known for its highly targeted nature. This means that it can only decrypt and execute on hosts with a certain hardware and software setup. Triggers can further be divided into time-based, remote-triggered (C&C), version numbered, or autonomous triggers on, e.g., successful target infection.

4.4.8 Cleanup

Malware authors may need to cover their tracks, for which a number of clean up functionalities have been developed, e.g., self-disinfecting removable drives [98], or self-removal after Time-to-Live (TTL) has expired. Concealing valuable technology by obfuscating or removing parts of the malicious code can significantly impede forensic analysis, as was seen in the case of Gödel [13, 98], which remains encrypted to this day.

4.5 Three Types of Smart-Grid-Enabled Malware

In this section we propose three hypothetical malware superclasses that are optimally suited for smart grid attacks. They are intended as examples to serve utility companies and researchers in their efforts to develop and implement proactive and reactive security measures for critical infrastructures. We are, however, aware that this information could be used for crafting next-generation malware.

Our classification relies on three main sources. We first consider analogies to the works of Li et al. [114], which proposed the establishment of a Cyber Center for Disease Control. We then borrow definitions from the Centers for Disease Control and Prevention (CDC) [22] as a main reference in order to draw analogy between the spreading of malware in cyberspace and the spreading of diseases in the real world. Lastly, we draw from Staniford et al. [166], who propose the "contagion" model that hardly leaves a trace for detection mechanisms to pick up on.

4.5.1 Pandemic Malware

A "pandemic" is defined [22] as the rapid spreading of a disease across an extensive geographic space, usually affecting many individuals. We draw an analogy from this example to the ICT realm and reuse this term to refer to the first category of smart grid malware. This type of malware is generally rapidly spreading, with the objective of aggressive propagation to as many victims as possible. The main goal of pandemic malware is the almost immediate infection of large areas of the network before countermeasures can mitigate it.

According to our investigations in Section 8.1, aggressive and noisy malware were preferred in the past, but recent developments have shifted towards stealthy, highly complex and modular types. Although techniques of rapid spreading techniques are not obsolete, modern detection mechanisms can today better identify and counteract them. Strong security implementations likely leave little attack surface for such malware types. Monocultures, however, increase propagation speeds and could, therefore, encourage a new generation of aggressive malware. In combination with hypothetical widely available zero-day vulnerabilities, such aggressive malware harbors the potential to spread quickly. This could have catastrophic consequences if critical infrastructures are affected. Zero-days may, however, be too expensive for malware of such a low sophistication level. Yet, delayed updating processes of recently discovered vulnerabilities typically leave a window of opportunity for pandemic malware, which may not be beyond the financial capabilities of attackers that can craft such malware. It may, however, be acquired on the black market [134].

Characteristics native to pandemic malware types include noisy scanning methods such as blind scanning or topological scanning, cf. Section 4.4.2, as well as simple payload construction, e.g., monomorphism, cf. Section 4.4.4. The evolution of malware shows that such aggressive types are often optimized for speed and do not implement much if any modularity, decreasing its overall complexity. We consider pandemic malware to be viable in the Internet landscape with a number of widely deployed services, but less likely to be so in a smart grid. Although it may find its way into a smart grid control network through improperly configured security mechanisms, modern heuristic detection should be capable of detecting such attacks.

Due to its simple design, the payload size has a general tendency toward a small file size without modularization options and few on-board features. According to [23,31,59,114,131,147,147,158,166] the smallest file size recorded is 400 bytes. The largest file size in this set is 60 kb. Therefore, we choose 500 bytes for our pandemic malware model. Furthermore, we chose a scan rate of 100 scans per second to represent aggressive scanning behavior for this malware category.

References and Significant Features

Real-world malware exhibiting features relevant to pandemic malware include: Code Red 1 and 2 [23, 131, 147], Nimda [23, 166], Slammer [114, 147], or Conficker [31, 59, 158].

The following list summarizes the most significant features:

- High propagation speeds at the expense of stealthiness.
- Aggressive scanning methods, that follow a topological subnet scan.
- Self-carried payload type and a simple, monomorphic architecture.
- Few propagation vectors predominantly in homogeneous topologies.
- Simple or older vulnerabilities.
- Presumably optimized for extortion or disruption attacks as espionage requires much greater stealthiness.
- No modularity, low complexity, therefore, low investments.

Countermeasures

- Regular security updates prevent low-complexity attacks by closing known vulnerabilities.
- High analytic speed in modern reactive security measures are of relevance due to the extremely high speed of infection rates expected.
- Heuristic detection should suffice to contain such types.
- Building critical infrastructures as monocultures should be avoided.
- Strong attack-resilience measures help mitigate attacks.
- Fallback systems help mitigate attacks and prevent collapse.
- Emergency restoration methods help mitigate attacks.
- Strict firewall rules with application white-listing decreases the versatility of malicious environments.
- Network segmentation prevents propagation.
- Strict access management confines users to specific controlled environments.

4.5.2 Endemic Malware

The term "endemic" refers to the constant presence and prevalence of a disease in a population within a geographic area [22]. We use this term to name the category of stealthy, persistent malware. Endemic malware types are modular, polymorphic, and can have multiple propagation vectors, thus, carry additional versatility. Modularity provides stealthiness features, minimizing the footprint, depending on local conditions, and allows for a multitude of payload-types with various attack-goals. Even though modules suitable for smart grid devices have not yet been seen in the wild, one must assume they will exist in the near future, especially when considering developments such as PLC-Blaster [108,165]. Furthermore, modularity at a smart meter scale can, in part, overcome the challenges of targeting heterogeneous network topologies through the use of multiple modules.

Various encryption and code obfuscation methods, e.g., polymorphism, augment defensive features and various propagation vectors increase reach, especially across air-gaps. Most endemic malware types considered in Section 8.1 are capable of network scanning and propagation. Yet, they utilize discovery methods that do not produce an excessive amount of network noise, e.g., hitlist-, permutation-, or distributed scanning. They are, however, detectable by modern reactive defense measures, which should motivate utilities toward implementing anomaly detection.

Our samples exhibit many persistence mechanisms. They are, however, sophisticated in the sense that endemic malware is difficult to purge from a network. Examples are discussed in Table 28, where we argue that evasion of anti-malware-tools, memory residency, code obfuscation, multi-layer-encryption, and reinfection are state of the art features in this category.

Endemic malware has a complex payload design with a general tendency toward large file sizes, although much of it comes from numerous on-board features that can be added in modular extensions. According to [13,31,56–58,70,95–97,123,169,173,175] the smallest file size in this set is 500 byte. The largest file size is 20 MB including hundreds of modules. We choose 5000 bytes for our endemic malware model to represent increased capabilities compared to pandemic malware's small payload (500 byte). Furthermore, we chose a scan rate of 1 scan per second to represent its less conspicuous scanning behavior.

References and Significant Features

This malware type represents generally stealthy, modular and persistent malware, such as Sality [31,175], Regin [169], Aurora [70,95,123], Stuxnet

[58, 123], Duqu 1 and 2 [13, 96, 173], Flame [13, 97], BlackEnergy3 [56], or CozyDuke [57].

The following list summarizes its most significant features:

- Sacrifice of propagation speed for increased stealthiness.
- Highly developed modularity allowing for a smaller footprint and adding some stealthiness.
- Polymorphism, which increases the stealthiness through code obfuscation and encryption.
- Multiple scanning methods allowing less conspicuous discovery.
- Multi-vector propagation (using zero-day vulnerabilities).
- Sophisticated persistence mechanisms such as detection-evasion, code-injection, or memory-residency.

Countermeasures

This malware type presumes all defense measures included in the pandemic model, supplemented by the following:

- Reactive measures, e.g., anomaly based intrusion detection and event correlation, that avoid the drawbacks of heuristic detection.
- Permanent network segmentation and strict firewall rules to prevent straight forward propagation.
- Content filtering can prevent host infections via, for instance, watering hole attacks or other drive-by downloads.
- Social engineering education to prevent unwanted access.

4.5.3 Contagion Malware

The contagion malware type builds upon the concepts of Staniford et al. [166] and represents malware that is especially difficult to discover. It propagates in a manner that is difficult or not at all traceable by network-detection mechanisms. Such methods may include either offline methods, e.g., removable drives, or highly obfuscated network channels, e.g., hidden communication that appends on legitimate traffic in either direction of the data flow. There are few suitable real world examples. For this reason, we include all malware types that are notoriously hard to detect.

Contagion malware has a very complex payload design with a general tendency toward large file sizes, similar to endemic malware. Therefore, it can utilize many modules and is capable of highly versatile application. According to [13, 72, 89, 95, 98, 99] the smallest file size recorded is 500 byte. The largest file size in our set is 2 MB, however, including all modules, thus, includes many modules we do not consider suitable for smart grid attacks. We choose 5000 bytes for our contagion malware category to represent increased capabilities compared to pandemic malware's small payload.

References and Significant Features

The following list summarizes the most significant features of contagion malware. Some features are recycled from the endemic class, however, contagion malware is taking stealthiness and persistence to a new level. Real world examples include Gauss [13,98,99], Equation [95,99], and AdWind [72,89,99].

The main features of contagion malware include:

- Highly developed modularity adding stealthiness and minimizing the footprint on networks and hosts.
- Multiple covert network scanning methods or captured network-information from the infected host.
- Multi-vector hidden propagation via zero-day vulnerabilities in an embedded payload-type.
- Network propagation via hidden channels or non-network channels, e.g., removable drives.
- Sacrifice of even more speed for increased stealthiness compared to endemic malware.
- Metamorphism and therefore highly sophisticated stealth features, encryption, obfuscation, and recompilation, further increasing persistence.
- Other sophisticated persistence mechanisms, including detection evasion, memory residency, service injection, or cleanup mechanisms.
- Firmware infection for added persistence against detection, even beyond host system recovery.

Countermeasures

This list presumes all measures included in the pandemic and endemic models, supplemented by the following:

- Anomaly detection and event correlation extend across multiple segregated networks.
- Permanent network segmentation with respect to the type of service used alongside with anomaly detection, i.e., only specific services may run inside smart grid VPNs.

4.5.4 Comparison of Smart Grid Enabled Malware

Unlike the pandemic model, the endemic and contagion models share one common characteristic. They are hard to detect, which gives them a clear advantage in stealthiness and persistence at the cost of speed. Table 12 applies the metrics from Section 8.1 onto the newly introduced smart grid malware types and lists countermeasures discussed in the aforementioned sections.

4.5.5 Illustration of a Threat Matrix for Smart Grid Enabled Malware

We illustrate a threat matrix, cf. Figure 16, to clarify the capabilities of different malware types, as presented in [47]. The figure depicts features, characteristics, capabilities, and particular strengths of the different malware types. The more sophisticated a malware feature, the more distant the corresponding point is located from the diagram origin. Therefore, assuming equal weighting for all features, a larger area represents a greater threat to defenders.

General features, e.g., the development effort shows that pandemic malware is simple, therefore, accessible to a larger group of attackers compared to the more advanced malware types. The source code of some variants [134] being accessible on the Internet as a template, even less skilled attackers can modify and implement their own version.

Endemic and contagion malware require increased resources in terms of development effort that may be a drawback for the attacker, thus, a benefit for defenders. However, this increased effort coincides with advanced on-board defense features against detection that are available in modular extensions, and represents a benefit for attackers in terms of improved attack capabilities.

Increased stealthiness features, which support reduced network scanning and stealthy malware propagation in networks, on one hand decrease propagation speed and frequently coincide with increased development effort. On

Table 12: Three types of smart grid enabled malware - Significant features and minimum countermeasures, as presented in [48]

	Metric	Pandemic	Endemic	Contagion
	Complexity / effort	Low	Medium	High
Propagation	Speed	High	Medium	Low
	Scope	Global	Targeted	Targeted
	Payload	Self-carried	Second-channel	Embedded
	Vectors	Few	Many	Many
	Modularity	None	Modular	Modular
Detection	Scanning	Aggressive	Stealth	None
	Payload transmission	Any	Stealth	Covert
	C&C [4]	None	Stealth [1]	Covert
	Morphism	Monomorph	Polymorph	Metamorph
On-board defense	Evasion of anti-malware tools	–	✓	✓
	Memory resident	–	~	✓
	Obfuscation	–	✓	✓
	Encryption	–	✓	✓[2]
	Service injection	–	✓	✓
	Reinfection	–	✓	✓
	Cleanup / uninstall	–	–	✓
Target	Unpatched Vulnerabilities	✓	✓	✓
	Zero-day	–	✓	✓
Minimum effective countermeasures	Security updates	✓	✓	✓
	Heuristic detection	✓	✓	✓
	Avoid monocultures	✓	✓	✓
	Resilience measures [3]	✓	✓	✓
	Fallback systems	✓	✓	✓
	Emergency restore	✓	✓	✓
	Anomaly detecion	✓	✓	✓
	Strict firewall rules	✓	✓	✓
	Access management	✓	✓	✓
	Content filtering	–	✓	✓
	Social engineering education	–	✓	✓
	Network segmentation & event correlation	–	✓	✓
	Network segmentation to type of service	–	–	✓

Notation: (✓) Yes, (~) Maybe, (–) No, (1) Encryption and obfuscation, (2) Encryption and recompilation, (3) E.g., redundancies, secure topologies or fallback-strategies, (4) Not simulated.

the other hand, advanced on-board capabilities enable the malware to optimize resource consumption in the network and/or on the host, supporting advanced features, e.g., additional attack vectors, obfuscation capabilities, or advanced scanning strategies. Figure 16 differentiates between general features, network domain features, and host domain features.

An additional malware categorization is with respect to pre-infection, the initial propagation phase, vs. post-infection, the operational phase. Pre-infection includes all actions that happen in the first few moments of a malware lifetime. This is the instant in time when the malware must propagate itself autonomously in a network. This also is the time frame we simulate in Section 9.

The operational phase, however, represents the malware's capability to stay hidden (unobserved) and persistent for an extended period. This includes low CPU usage by the malware such that defending software may not identify CPU overload for a system that is supposed to operate within certain specifications. Failure of the malware to do so opens opportunities for defending software that can detect either processes on the host that act suspiciously or excess network traffic for malware C&C or propagation activities. However, we discuss them with the defensive measures listed in Section 10.

Other host based features that correlate well with the development effort include the malware's payload structure, as discussed in Section 4.4.4. A monomorphic payload represents a simple construct that may change in size but produces similar signatures. Therefore, it can be detected reliably whenever heuristic signatures are available. A polymorphic payload complicates detection by scrambling its shape and size through encryption. Still, decrypted payloads will produce identical signatures on the local drive of the host, being detectable by heuristic methods. Malware featuring metamorphic payload requires the higher development effort, varying in size, shape, encoding, and encryption. Moreover, recompilation on the host system can be used to obfuscate any trace of the payloads presence, cf. [89].

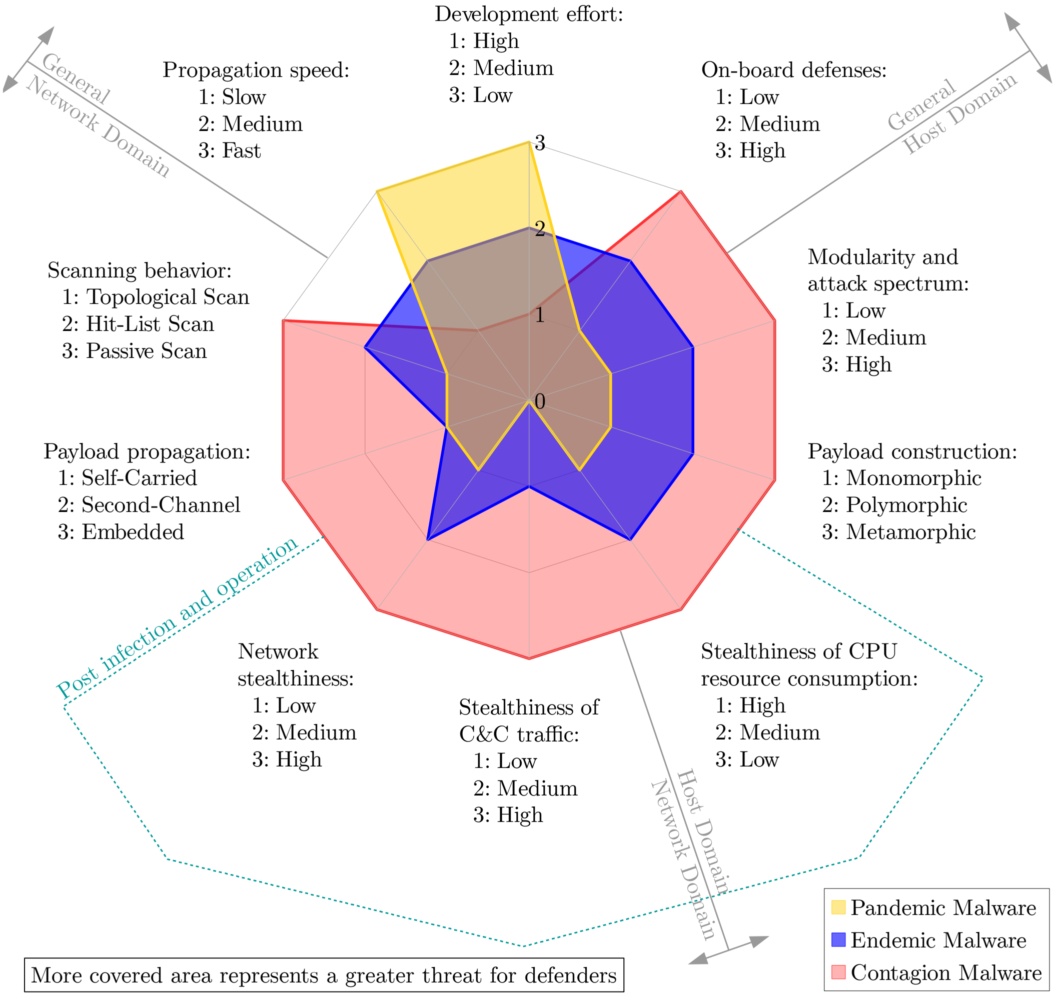


Figure 16: Illustration of a threat matrix of smart grid enabled malware, as presented in [47]

4.6 Vulnerability Abstraction

For our simulations we assume that the smart grid infrastructure has not been updated for a time, leaving the security level unpatched. We make the following assumptions in terms of vulnerabilities:

- We generally assume zero day vulnerabilities in the application layer [83], thus, in the implementation of software on smart grid devices including the field nodes, smart meters, control equipment, gateways, industrial computers, servers, and other equipment.
- These vulnerabilities are assumed to affect all node types, as can occur in monocultures of devices, cf. [106, 116], allowing remote code execution and administrator rights (privilege escalation) on all host systems. Therefore, infected hosts can appear to be any other type of node toward victims (spoofing).
- We assume that all malware types (lesser and more capable types) can exploit the vulnerabilities, otherwise we cannot compare the three malware classes. Thus, advanced malware (endemic and contagion) use zero day vulnerabilities to propagate, while simple malware (pandemic) use known but unpatched vulnerabilities.
- We exclude vulnerabilities concerning well established and tested protocols and methods, e.g., IP, OLSR, and PLC. Therefore infections only commence upon direct communication between nodes, and not as drive-by infections along the way of routing or switching nodes. This results in no infection of hop-nodes in mesh networks which only pass packets on. See Figure 17 for an illustration of the infection path from a LL-node (origin) to the gateway (target).
- We assume a layer 2 vulnerability [83] in one simulation model, namely the decentralized topology. Section 9.3.4 elaborates on the network structure. Therefore, infected nodes can directly connect to nodes in a neighboring network, thus, jump into another mesh network, without going through legitimate channels, i.e., the gateway and backhaul infrastructure. However, all other topologies follow a strict segmentation between different sub-networks. We introduce this exception due to limitations of the OLSR protocol whereby we cannot simulate one large mesh network with hundreds of nodes. Since the decentralized topology is altered from the theoretical model, cf. Section 3.4.4, we have to model it with a backhaul infrastructure and sub-meshes.

5 Malware Propagation Simulation Model

Notice of adoption from previous publications:
Parts of the contents in this section have been submitted for publication in [46]. We introduce metrics on evaluating malware features, such as propagation, scanning, or infection ratio, among others. The authors main contribution was the development of the metrics and the malware models. The co-authors contributed additional benefit to testing, overhauling the metrics, plausibility checks, and the limitations of those metrics.

This section includes the basics for our simulation model. We outline different technological abstractions, the models capabilities, its performance, and all basic assumptions relevant to the simulation environment. Our simulation model is based on the ns3 simulation environment [137]. It was developed by a collaboration of the author and a student working on the diploma thesis (work in progress) as proof-of-concept on malware attack simulations. The author of this work expanded the simulation model upon additional network topologies and malware features.

5.1 Methodology

Our approach for achieving results that satisfy our research questions, cf. Section 1.3, uses our theoretical malware models published in [39,44,48], the network topologies from [44], and the simulations that support those theoretical works, cf. [46]. We use ns3 [137] which supplies many basic models, however, we modify some of these models for our malware and topology implementations. ns3 is an open source platform and based on C++, thus, our models could be reused by other researchers. We utilize network simulations because of their scalability and reproducibility. Additionally, the development of real malware implementations operating in existing smart grids are not an option for security reasons. Furthermore, there are no fully equipped smart grids available for testing yet.

Our goal is to show what could, in future, become reality when smart grids are widely deployed. Aside from theoretical considerations and simulations, there are alternative methods that could be used for researching malware in smart grids. Mathematical models could be developed to formulate the progression of malware propagation, that represent our generic malware models. For this reason we present calculation metrics, cf. [46], that cover most aspects on, e.g., malware scanning, propagation, and infection ratios, that can be used in theoretical models as well as in simulation. Furthermore, emulation in a hardware in the loop approach could provide enclosed environments for testing real hardware with malware implementations. However, these test farms would have to be secured well, because if real implementations of those attack technologies are being developed, one must also invest in vulnerabilities of existing devices.

5.2 Technological Abstraction

This section includes the technological abstractions of communication technologies. We establish our simulation environment by describing the link technologies used and define the behavior of hosts, and malware.

- The LL-nodes, i.e., smart building managers, and the ML-nodes, i.e,. regional control nodes, are allowed to act autonomously and push data or begin sending on periodic pull request from higher level nodes, as was described in Section 3.4.
- We assume infected nodes can spoof other types of nodes, e.g., gateways, expanding their capabilities to operating modes they should otherwise not posses. Furthermore, malware can change its host-behavior to imitate legitimate ML-nodes and contact other nodes, e.g., fake update servers, or fake control nodes.

5.2.1 Wired Link Technologies

We discussed established communication technologies that can be used in smart grids in Section 2.2. Table 3 elaborates on coverage areas and scope. The following list include arguments for selecting certain link technologies and exclude others. Furthermore, we discuss the technological abstractions of our model. We limit our model to optical fiber and PLC technologies that are used as example technologies because of their characteristics and their likelihood for future deployments.

- We assume that those topologies that use wired links between HL-nodes and ML-nodes are based on optical fiber. According to WSTW [184] maintenance and construction work on the power grid is generally used to lay down spare tubes or optical fiber cables parallel to the power lines during construction work. The reason for using optical fiber is that magnetic fields induced by the power lines cannot influence the communications link. We assume WDMA over multi-mode optical links as described in [130] as the established technology.
- The maximum optical link transmission speed is limited to 40 GBps over a maximum link length of 800 meters between ML-nodes and HL-nodes. [119, 184]

- We assume that the wired links between LL-nodes and ML-nodes are based on PLC. Due to the LL-ICT network size, we consider it unfeasible connecting every household with a dedicated optical link in the near future.
- Although, commercial DSL networks are widely available, we restrict our choice of link technology to dedicated lines under the control of the network operator due to security considerations. Therefore, we do not operate this critical infrastructure over external networks. We utilize PLC and FDMA, which delivers according to [107, 156, 187] good usability and high scalability up to distances of 1000 m, while retaining tractability on multiple branches connecting to one point.
- The maximum PLC transmission speed is limited to 1 Mbps over a maximum link length of 1000 meters between LL-nodes and ML-nodes. [74, 107, 156]

5.2.2 Wireless Link Technologies

We discussed established communication technologies that can be used in smart grids in Section 2.2 and Table 4. However, we limit our simulation model to WiFi and wireless M-Bus based mesh networking. Therefore, we do not consider satellite, Z-Wave, WPAN, and WiMax. Additionally, we exclude the use of mobile carrier networks (3G, 4G, 5G) from our simulation model because critical switching operations require emergency bandwidth, that cannot be guaranteed in external networks that are not within the control of the utility company. Furthermore, 3G is known for its insecurity as discussed in [88]. We are not convinced that sharing an external media with a commercial provider meets the security requirements necessary for power switching operations, cf. defense measures in Section 10. The following list summarizes the assumptions on wireless technologies:

- Wireless communication is modeled with an OLSR based radio mesh, owned and operated by the utility company. However, we assume this communication channel is End-to-End (E2E) encrypted and not shared for commercial purposes due to security considerations, thus, we simulate a dedicated infrastructure.
- The main considerations to prefer the OLSR protocol over other mesh protocols is its prevalence as a renowned standard in mesh networking among proactive protocols. Being a proactive protocol, each route through the network is calculated before hand, which is beneficial in terms of delay performance. Since we utilize an extremely static topology proactive protocols are beneficial over reactive protocols, i.e., lower control message requirements.

- There are several scenarios, cf. Section 5.5, where nodes communicate via OLSR mesh network. We assume that the necessary bandwidths will be achievable inside the 2.4 GHz and 5 GHz IEEE 802.11 bands. Additionally, a cognitive radio approach including the 440 MHz licensed radio band provides stable bandwidth across several frequencies, cf. Table 4, would be possible. According to Kamstrup [91] the 440 MHz band is capable of up to 500 m sending distance in urban areas, at a reduced bandwidth of 1 Mbps and can, therefore, accommodate smart metering requirements over larger distances.

- The OLSR protocol requires a minimum of 2 cycles multipoint relay (MPR) selection until all nodes are known and initial control traffic has settled, as discussed in [10, 85, 121]. Therefore, we allow three full cycles before starting any simulation, to minimize the OLSR impact, thus, we simulate established networks.

- We conduct initial simulations, cf. Section 9.1.3, on the emergence of control messages and how they impact the network. Although we limit our maximum network size to 70 nodes, we also limit the OLSR protocol for longer control-message cycles. According to [10, 143, 168], the OLSR control message interval can be increased for stationary networks. This is due to the limited scaling capabilities in dense and large networks, with a changing topology. We, however, use an extremely static topology. Therefore, OLSR does not need to converge control messages every 5 seconds as per the standard, cf. [139]. After discussion with the ns3 OLSR-model developers, we conclude to set the interval to 500 seconds instead of 5 seconds for all simulations because our node locations never change.

- The OLSR-mesh link bandwidth decreases according to the ns3 physical model which is based on wireless signal propagation, therefore, dependent on the link distance.

5.2.3 Link Technologies Comparison

Figure 17 illustrates our network abstractions. All nodes of the same type are identical, however the link technologies between them vary. While mesh networks allow benefits in terms of resilience to node failure or links, they are also expected to provide those same benefits for attackers that can utilize alternative routes to infect victims. We illustrate this with many available connections that could be used by, e.g., the least costly connection for legitimate traffic, and optimized malware propagation. Wired links on the other hand are confined to strict routes, that are predefined by the power grid topology, cf. Section 3.5, providing better control over the data flow. Fur-

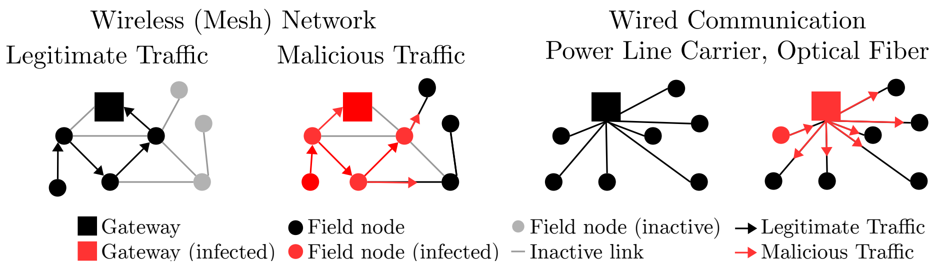


Figure 17: Topology Abstraction

Table 13: Advantages and disadvantages of link technologies

Technology	Key benefit	Key drawback	Scope
Optical-Fiber	High transmission rates over long distances	High cost, needs fitting during construction.	WAN - MAN
PLC	Utilizes the existing power infrastructure	Limited crosstalk capability, limited distances	MAN - NAN
WiFi Mesh or M-Bus	No need for excavation, no maintenance of cables	May require antennas outside the buildings	NAN - HAN

thermore, we assume WDMA and FDMA for wired connections, cf. Table 14, thus, resulting in P2P connections.

Table 13 provides an overview of the key benefits and drawbacks of all link technologies we use in our model. The OLSR-mesh speed [91] is generally lower than optical fiber speeds [119] but faster than low bandwidth PLC links [107, 156, 187], cf. Table 14. The wireless link speed depends on the distance between nodes by the physical propagation model.

Table 14 summarizes the link technology, link speeds and transmission modes used in the simulation model.

Table 14: Technical Data Comparison of Link Technologies

Technology	Link Speed [Mbps]	Mode	Reference
Optical Fiber	40000	WDMA	[119]
PLC	1	FDMA	[107, 156, 187]
WiFi Mesh	11 (Physical propagation model)	OLSR	[91]

5.3 Capabilities of the Simulation Environment

We defined the goal of this work, cf. Section 1.3, to develop effective detection metrics and countermeasures. For this we develop a simulation environment that can simulate three types of malware in four types of network topologies. All aspects, the malware behavior, the network topologies, and the evaluation metrics have been optimized for smart grid implementation and presents extrapolated possible future developments. We developed our simulation models with future works in mind and included a random seed function that can be turned on, affecting three parameters; The start time of native traffic, the initial target selection of patient zero in the propagation function, and a random time interval added to the exploit time on each host. We, however, aim to achieve *reproducibility* for all simulations, thus, we are not iterating each parameter in hundreds of simulations. Therefore, as discussed in Section 9.1.2, large scale statistical analyses are out of scope. However, we touch on the issue in the improvement section, cf. Section 11.7. Although we include several tests that outline the simulation environment, cf. Sections 5.3.1 and 5.3.2, the random seed function is limited by the following:

- The random start condition of the native traffic is translational invariant, only shifting the native traffic alongside the time axis, parallel to malicious traffic.
- Random target selection (patient zero) behaves isomorphic, as iterating every node leads to arrival at the starting point.
- The host infection (exploit) time includes a small random component (< 1 ms). However, we minimize the exploit time on all hosts because it is out of scope for detailed network analysis, thus is negligible, and we deactivate the random component.

5.3.1 The Effect of Increasing the Wireless Distances

Initial simulations conducted in wireless mesh networks to set our basic simulation environment show that a wireless distance of 0 and 200 meters between the nodes has no significant effect on the propagation of malware. Section 9.1.1 elaborates on these results. This is true up to a maximum network size of 80 nodes. However, in networks with more than 80 nodes the increasing number of nodes influences the required control traffic, thus, the link bandwidth is consumed in the process. According to Palma et al. [143] and Audeh [7] large mesh networks degrade in performance with increasing node numbers. The control traffic on mesh networks does not scale indefinitely, therefore, increases at the cost of bandwidth. This is true for most mesh network protocols, unless they operate with a hybrid infrastructure including wired backhaul links alongside with the wireless mesh and relieving control- and routing traffic from the mesh.

Therefore, we limit our approach to a maximum network size of 80 nodes per mesh, cf. [7, 143] and Section 9.1.2, leaving us with a hybrid infrastructure that consists of mesh networks connected via wired backhaul links for the cell- and mesh topology. Additionally, we interconnect all mesh networks via bridge nodes, i.e., nodes that are located at the border of a mesh network and have an additional network interface for neighboring networks to overcome this pitfall of mesh technology with the decentralized topology. These limitations lead to partial loss of network resilience against node failure, however, we do not consider node failure in our simulations. The simulation time generally commences until full infection is achieved or the simulation concludes that all nodes have been tried for infection. Furthermore, at a maximum mesh size of 80 nodes, cf. Section 9.1.1, the wireless distance between the nodes has no significant impact, reducing the complexity of our model. Therefore, we generally limit the simulations to a node distance of 108 meters, i.e., the mean value shown in Figure 14(b) for our statistical abstractions of the ML-nodes. This distance seems reasonable since each simulated node represents 12 apartments in a city district, as elaborated in Section 3.5.2.

5.3.2 Simulation Performance of the Mesh Network Models

In this section we discuss the performance of the mesh-based simulation environment and its maximum total simulation size. This section concerns the basic setup of the mesh topology and the decentralized topology because we simulate many mesh nodes. The other topologies may utilize mesh technology in a reduced capacity, thus, are within the maximum amount of nodes. Section 9.1.2 includes detailed simulations and figures that illustrate the performance of the overall network model according to large scale simulations up to 3000 nodes. We initially had the suspicion that 3000 nodes i.e., necessary to accommodate the parent data set, cf. Section 3.5, may not converge in one single network. Therefore, we split the mesh networks into sub-meshes and use a backhaul infrastructure between gateway nodes. This hybrid infrastructure allows us to accommodate a greater number of nodes over several sub-networks.

Additionally, we discovered that the simulation environment can be reproduced several times and represent a much larger set of nodes. Due to the clustered nature of the sub-mesh-based approach, we assume that for instance another set of (copy of) all node types represents a different city district of the same structural makeup. Therefore, we setup our model to represent on section of the city, instead of all, to provide results with a reduced number of total nodes, relieving computational effort. The simulation model cannot be split over multi-core clusters, due to the ns3 mesh-model, that is restricted to multi-core precessing only in P2P networks. We test several scenarios that allow us to reduce the size of the simulation environment, cf. Section 9.1.2. Therefore, we set the largest overall network size to 576 nodes, with 64 nodes per mesh network and 9 mesh networks, cf. Section 5.5. We summarize these changes in Table 16.

5.4 Simulation Topology

We base our simulation models on the theoretical models introduced in Section 3.4, cf. Figure 7, and the parent data set available in Section 3.5.1. The simulation topologies below elaborate on the parameters harmonized from the parent data set, summarized in Table 16.

5.4.1 Centralized Topology

The centralized topology, cf. Figures 7.a and 18, includes 31 LL-nodes per ML-node, 18 ML-nodes, and 1 HL-node. All nodes are connected in star topology to the next hierarchy level and all communication links are wired in a dedicated infrastructure. The ML-nodes are connected to the HL-node via optical fiber links. The LL-nodes are connected to the ML-nodes via PLC.

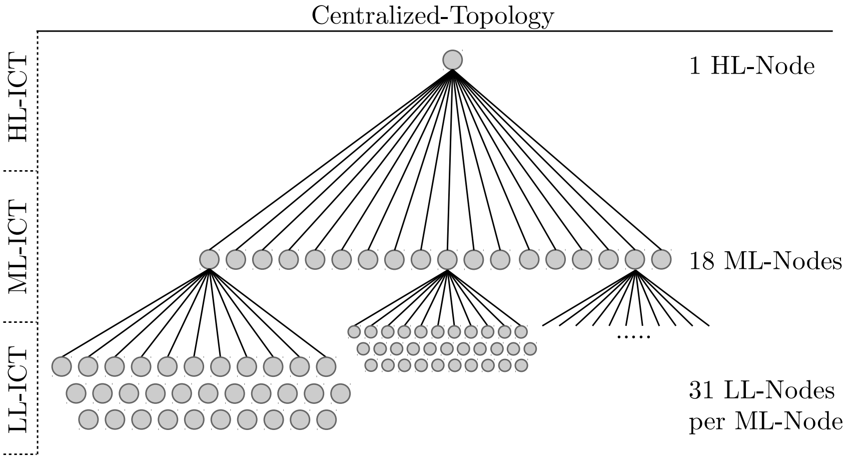


Figure 18: Simulation Topology - Centralized Configuration

5.4.2 Cell Topology

The cell topology, cf. Figures 7.b and 19, includes 31 LL-nodes per ML-node, 18 ML-nodes, and 1 HL-node. The LL-nodes are connected in star topology to the ML-nodes, thus, there are 18 low level sub-networks each with a dedicated ML-node as its master. All ML-nodes are connected via a wireless mesh network, while several uplinks (every 6th ML-node) connect them to the HL-node. The reason for using uplinks from selected ML-nodes is that those are connected to the optical fiber network, representing the closest possible approximation to the existing topology, cf. Section 3.5.1.

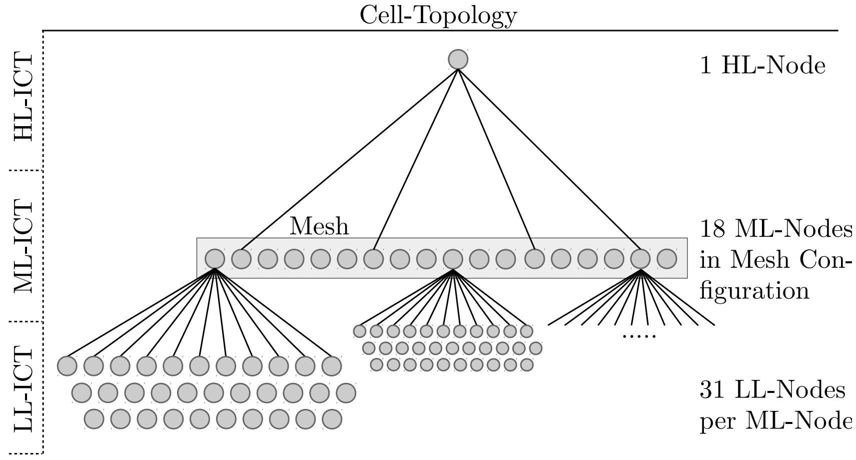


Figure 19: Simulation Topology - Cell Configuration

5.4.3 Mesh Topology

The mesh topology, cf. Figures 7.c and 20, includes 63 LL-nodes per ML-node, 8 ML-nodes, and 1 central HL-node, whereas the HL-node aside from the backhaul infrastructure also manages one sub-mesh network. Therefore, all 9 sub-meshes have a dedicated gateway (ML-node or HL-node) and are connected via a backhaul infrastructure. Since the backhaul infrastructure is not immune to the attack, lateral infection is possible when gateway nodes are infected.

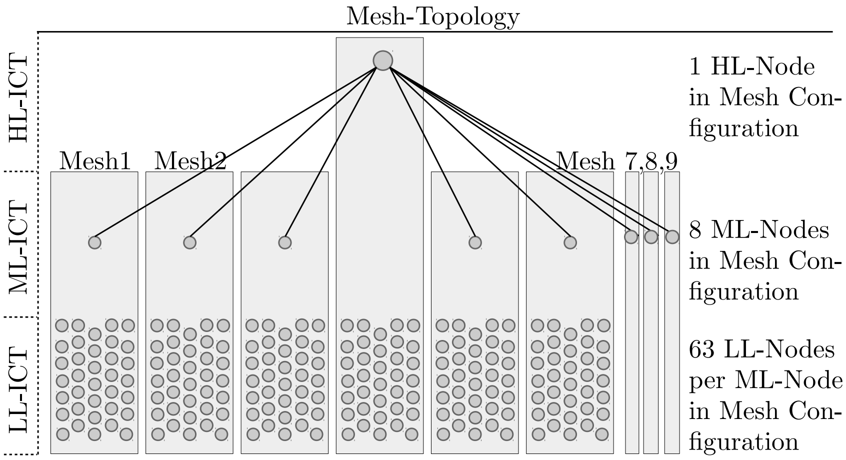

Mesh-Topology
HL-ICT
ML-ICT
LL-ICT
Mesh1
Mesh2
Mesh 7,8,9
1 HL-Node in Mesh Configuration
8 ML-Nodes in Mesh Configuration
63 LL-Nodes per ML-Node in Mesh Configuration

Figure 20: Simulation Topology - Mesh Configuration

5.4.4 Decentralized Topology

The decentralized topology illustrated in Figures 7.d and 21 includes 63 LL-nodes per ML-node, 9 ML-nodes, and no central HL-node. All nodes are organized in 9 mesh networks which are connected via a backhaul infrastructure as elaborated in Section 5.4.3.

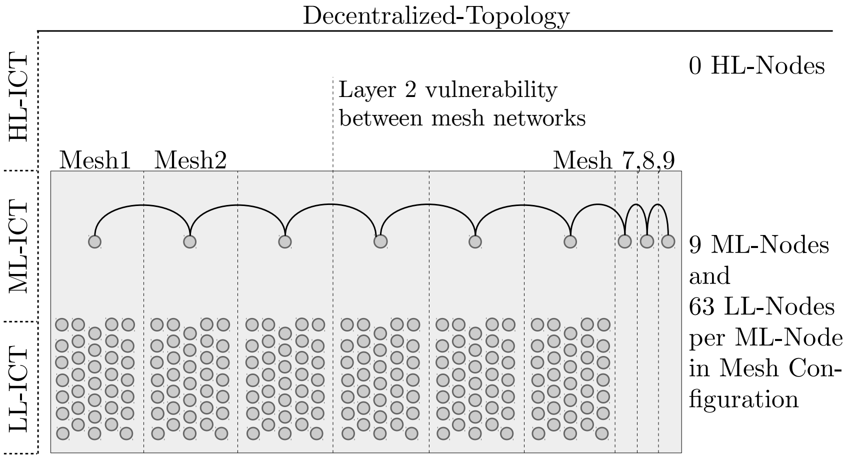

Decentralized-Topology
HL-ICT
ML-ICT
LL-ICT
0 HL-Nodes
Layer 2 vulnerability between mesh networks
Mesh1
Mesh2
Mesh 7,8,9
9 ML-Nodes and 63 LL-Nodes per ML-Node in Mesh Configuration

Figure 21: Simulation Topology - Decentralized Configuration

We do not simulate one large mesh network as introduced in our theoretic model (Section 3.4.4) due to limitations with decreasing performance at increasing node numbers (Section 9.1.2). Therefore, we setup the decentralized topology with a sub-network approach and connect them via a backhaul infrastructure. Furthermore, we connect the subnetworks directly via border nodes that can communicate also with the neighbor network, that allows infecting nodes that are in radio range regardless of sub-network segmentation. This vulnerability is giving attackers the capability to move laterally even without the backhaul infrastructure. We allow the malware model to prioritize this lateral movement vulnerability, cf. Section 4.6, over the application layer vulnerability that is used to move over the backhaul infrastructure.

5.5 Scenarios

The smart grid specific malware types introduced in Section 4.5 are simulated across the topologies introduced in Section 3.4. Table 15 maps the scenarios to all topology types and malware types.

Table 15: Scenario overview

Scenario name and number		Topology	Malware
0.1	Maximum mesh-size, cf. Section 9.1.2	Decentralized [1]	Endemic
0.2	Maximum overall network size, cf. Section 9.1.2	Decentralized [1]	Endemic
0.3	Increased OLSR message cycle, cf. Section 9.1.3	Decentralized [1]	Endemic
0.4	Network segmentation and Monocultures, cf. section 9.2	Decentralized [1]	Endemic
1.1	Centralized pandemic, cf. Section 9.3.1	Centralized	Pandemic
1.2	Centralized endemic, cf. Section 9.3.1	Centralized	Endemic
1.3	Centralized contagion, cf. Section 9.3.1	Centralized	Contagion
2.1	Cell topology pandemic, cf. Section 9.3.2	Cell	Pandemic
2.2	Cell topology endemic, cf. Section 9.3.2	Cell	Endemic
2.3	Cell topology contagion, cf. Section 9.3.2	Cell	Contagion
3.1	Mesh topology pandemic, cf. Section 9.3.3	Mesh	Pandemic
3.2	Mesh topology endemic, cf. Section 9.3.3	Mesh	Endemic
3.3	Mesh topology contagion, cf. Section 9.3.3	Mesh	Contagion
4.1	Decentralized pandemic, cf. Section 9.3.4	Decentralized	Pandemic
4.2	Decentralized endemic, cf. Section 9.3.4	Decentralized	Endemic
4.3	Decentralized contagion, cf. Section 9.3.4	Decentralized	Contagion

Notation: (1) Mesh clusters with a large number of nodes represent our most challenging test-case.

Table 16 elaborates on node and link numbers and their connection type for each topology used in the simulations.

Table 16: Topology settings for main scenarios in detail

	Centralized	Cell	Mesh	Decentralized
HL-nodes [a]	1	1 [b]	1 [b]	0
ML uplink	18	3 [c]	8	0
ML-nodes / gateways	18	18 [c]	8	9 [d]
LL uplink	31 [e]	31 [e]	Mesh [f]	Mesh [f]
LL-nodes	31 [e]	31 [e]	63 [g]	63 [g]
Total nodes	577	577	576	576

Legend:
(a) Manage the backhaul infrastructure.
(b) Manages one sub-mesh.
(c) Every sixth ML-node has an uplink, cf. Table 8.
(d) Each mesh network connects to others via a backhaul infrastructure.
(e) Per ML-node.
(f) Each sub-mesh connects to the backhaul infra. via its gateway.
(g) Per mesh network.

5.6 Parameters

Table 17 elaborates on all parameters, i.e., input variables, and all levels, i.e., values of parameters, used in our simulation model.

Table 17: Parameters Summary

Parameters	Level	Reference
Random seed	Random [1]	Section 5.3
Topology type	4 types	Section 3.4
Number of HL-nodes	1, depends on topology	Section 3.4
Number of ML-nodes	31, depends on topology	Section 9.1.2
Number of LL-nodes	567, depends on topology	Section 9.1.2
Number of dwellings	12 per LL-node	Section 3.5.2
Sending Cycle ML link	100 [kB/s]	Section 6
Sending Cycle LL link	100 [kB/60s]	Section 6
Wired distances	200 [m]	Section 3.6.1
Wireless distance a	100 [m] in mesh and decentralized topology	Section 3.6.2
Wireless distance b	400 [m] in cell topology	Section 3.6.2
Link speed optical fiber	40 [Gbps]	[119]
Link speed PLC	1 [Mbps]	[107, 156, 187]
Link speed Wifi / M-Bus	Physical model [2]	[75, 91]
OLSR converge time	500 [s]	Section 5.3
Native traffic size	100 [kB] per node	[5, 6]
Malware type	3 Types, cf. Section 4.5	[48]
Scan rate pandemic	100 [1/s]	Section 4.5
Scan rate endemic	1 [1/s]	Section 4.5
Payload size pandemic	500 [B]	Section 4.5
Payload size endemic and contagion	5000 [B]	Section 4.5
Exploit time	0.001 [s]	Section 4.4.4
Patient zero node	Node 0 (LL-node)	Section 4.4.1
Infection start	400 [s]	Section 5.3

Notation:
(1) Used as a fixed value, but could be randomized in future works.
(2) Can be changed to another propagation model, IEEE 802.11b.
B = Byte
bps = bit per second
m = meter
s = seconds

5.7 Verification

We base our simulation model, cf. Section 5, on our theoretical investigations in Sections 3 through 4 as published in [44,48]. These works represent state-of-the-art research that include a wide range of network topologies including in depth evaluation of the most significant malware features. We isolate these features and topology metrics and reproduce them in the ns3 simulation environment with the support of tested standard models [137]. The malware attack model was developed by the author (concept, development, and programming) in cooperation with a student (programming) and based on the theoretical models introduced in [48]. The topological models were developed by the author and are based on [44].

5.8 Validation

We expect the validity of the simulation models to be similar to real world attacks as per the evaluation of existing malware types, cf. Section 8.1. Our results show that different malware types behave as expected from the state of the art. It also shows, e.g., significant speed benefits of certain malware types at the cost of stealthiness, and very slow propagation at heavily increased stealtiness, and the increasing trend toward better stealthiness and obfuscation features. We can confirm these results in our simulations, cf. Section 9.3, for all malware types that were modeled.

6 Malware-Attack Evaluation Metrics

Notice of adoption from previous publications:
Parts of the contents in this section have been previously published in [46]. We introduce metrics on evaluating malware detection, propagation, infection ratios, and attack-efficiencies. The authors main contribution was the development of the metrics and the malware models. The co-authors contributed additional benefit to testing, overhauling, plausibility checks, and their limitations.

According to research question 3, cf. Section 1.3, we elaborate on metrics that are used to evaluate the behavior of three malware types in four network topologies. They include, scanning stealthiness, scanning efficiency, propagation stealthiness, propagation efficiency, infection ratio, infection efficiency, infection duration, noise suppression efficiency, attack-efficiency, and attack-defendability. The data that is extracted from our simulations can be processed with these metrics in order to compare the malware behavior. All simulations follow our communication model, as illustrated in Figure 6. Additionally, we extended our calculable metrics upon theoretical metrics that extend our model upon potential support for future work, namely attack-containment and attack-resilience. They consider numerous defensive solutions that are not simulated. We include a full set of all notations used in these metrics in Table 21 at the end of this section.

6.1 Methodology

We develop these metrics to present calculable output for our simulation environment for comparison of different malware types in different network topologies, and to satisfy our research questions, cf. Section 1.3. They represent our basic evaluation. However, because of security reasons it is impossible for us to develop real malware which could run in existing smart grid environments of emulated hardware in the loop test benches. Our goal is to show what attack vectors could, in future, be possible when smart grids are widely deployed, focusing on the future development of malware.

6.2 Node Infection Ratio

We define the *infection ratio* (R_{inf}), i.e., the ratio of all nodes that are infected during the simulation, within the interval [0, 1], where zero means no node is infected and one means all nodes are infected. Generally, field nodes have one network interface. However, the gateway nodes, representing the connecting nodes between different networks, can have several network interfaces, one for each sub-network. They are not counted as separate interfaces per node in Formula 1, because each inbound connection that infects a node also infects all of its interfaces, cf. Section 3.3. Therefore, n_{host} and n_{inf} unify those network interfaces in one node. Furthermore, we define that each node may only operate one network interface per network, thus, nodes with several network interfaces must connect to as many networks as they have interfaces.

It is defined as follows:

Notation:
s = Number of sub-networks
n_{inf}(i) = Number of infected nodes in sub-network i
n_{host}(i) = Existing number of nodes in sub-network i
R_{inf} $\in$ [0, 1]

$$R_{inf} = \frac{\sum_{i=1}^{s} n_{inf}(i)}{\sum_{i=1}^{s} n_{host}(i)} \tag{1}$$

Consequently, the *ratio of clean nodes* (R_{clean}), i.e., the percentage of nodes that have not been infected, is defined as follows:

$$R_{clean} = 1 - R_{inf} \tag{2}$$

6.3 Infection Durations

We define five points in our simulation when significant events occur:

- We start our simulation, i.e., we set the simulation time to 0, when *patient zero* is infected manually. Patient zero is the result of the initial attack vector being successful which can include any number of attack types, e.g., physical access to the LL-node, lateral movement from local IoT devices, or lateral movement from an RTU in the enterprise network. Furthermore, we note that patient zero could be set to any node in our simulation model, however, decided to restrict it to LL-nodes for our simulations. The reason for this decision is that LL-nodes represent the most easily accessible nodes, the least protected nodes, and also those nodes that have the longest propagation path for a successful attack. Additional detail can be found in the improvement section 11.7.

- The next significant point in time is met when the *first gateway* ($T_{first.GW}$), i.e., the first ML-node is infected. The attacker now gains the capability to either spy on the aggregated data of this node or disable the communications and electricity supply to the entire sub-network controlled by this particular ML-node.

- The next point represents the infection of the *control center* ($T_{C.Center}$), i.e., the HL-node. Now the attacker has the power of shutting down all nodes. However, selective spying on nodes is not possible, except for the aggregated data that is sent to the control center.
- Similar to the infection of the control center, the infection of the *last gateway* ($T_{last.GW}$) provides the attacker with extensive power over all ML-nodes. This is also the last instance at which trustworthy measurements can be expected from any ML-node. At this stage the attacker can spy on all aggregated data or disable communications and electricity supply to all gateways, regardless if the field nodes are infected. Such an attack can destabilize the power grid as seen in the Ukraine attack [34, 56].
- Next, we define the point when *75% of all nodes are infected* ($T_{75\%.nodes}$), at which the attacker has access to a significant number of nodes. This includes LL-nodes, ML-nodes, and the HL-node. The chance that a specific target of interest is among them is high. Furthermore, it opens additional attack vectors such as selective deactivation of LL-nodes, selective data theft, building a botnet, or non-permitted distributed computing on a large number of infected nodes, e.g., Bitcoin mining.
- Finally, we introduce the point during the infection process that marks when all nodes have been infected ($T_{all.nodes}$), i.e. no more nodes can be infected. Therefore, the malware does not spread any further. However, it is possible that full infection cannot be achieved. The reasons being packet loss on the communications link or that nodes are not scanned by the malware algorithm, e.g., if errors exist in the hit-list. Therefore, we define $T_{last.node}$ as an alternative final point that represents all nodes known to the malware that have been tried. Furthermore, $T_{last.node}$, being smaller or equal to $T_{all.nodes}$ marks the end of a simulation.

Summarizing our simulation is characterized by the following time instances:

$T_{first.GW}$ = Time until first gateway is infected.
$T_{C.Center}$ = Time until the control center is infected.
$T_{last.GW}$ = Time until last gateway is infected.
$T_{75\%.nodes}$ = Time until 75% of all nodes are infected.
$T_{last.node}$ = Time until the last node is infected. $T_{last.node} \leq T_{all.nodes}$
$T_{all.nodes}$ = Time until all nodes on the network are infected.
$T \in [0, \infty]$

In Figure 22 we show an example infection graph with the infection ratio R_{inf} as a function of the infection time. Colored areas in the graph illustrate how fast countermeasures would need to apply in order to prevent the malware

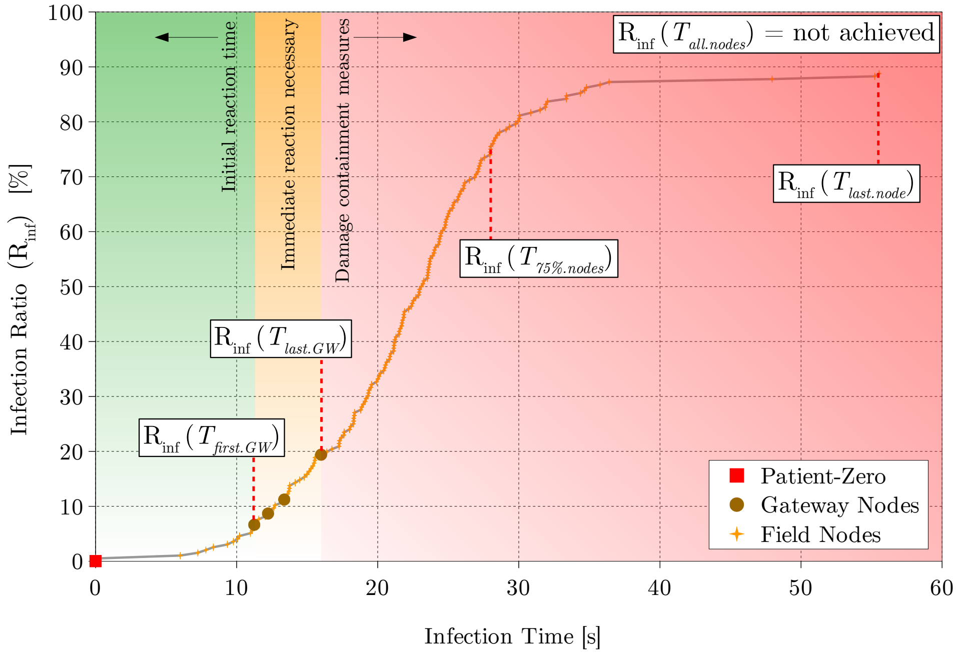


Figure 22: Infection graph with significant infection times

to reach the next level of infection defined by these infection durations.

6.4 Minimum Malware Infection Duration

The *minimum malware infection duration* ($T_{infect.min.}$) is a theoretical limit that represents the minimum duration that the infection of all critical nodes requires depending on different malware types and network topologies. We define $T_{infect.min.}$ as the sum of minimum packet transmission times and propagation delays for different link technologies, depending on the number of links required to reach all critical nodes in a particular topology. The transmission times and propagation delays are calculated for each link type, malware type, and network topology. However, we exclude local node processing time, queuing, congestions, and local traffic from this metric.

The packet transmission time is calculated by the payload size in bytes multiplied by 8 bit per byte divided by the theoretical link data rate bit per second, cf. Table 17. The propagation delay is calculated by link length, cf. Tables 9 and 11, divided by the propagation speed of wired and wireless links.

Table 18 summarizes the minimum packet transmission time and propagation delay for each link technology.

Table 18: Minimum packet transmission time and propagation delay

Delays [ms]	LL uplink	ML uplink	Mesh link
Pandemic ^{1} transmission delay	4	0.0001	0.3636
Endemic ^{2} transmission	40	0.0010	3.6364
Contagion ^{2} transmission	40	0.0010	3.6364
Pandemic propagation delay	0.0002	0.0010	0.0003
Endemic propagation delay	0.0002	0.0010	0.0003
Contagion propagation delay	0.0002	0.0010	0.0003

Notation: ms = miliseconds, (1) Payload size is 500 byte, (2) Payload size is 5000 byte

Table 19: Minimum malware infection time per required hop

Number of hops	Centralized	Cell	Mesh	Decentr.
Required ML uplinks	1	1	1	8
Required LL uplinks	1	1	n.a.	n.a.
Required mesh links	n.a.	1	4	4
$T_{infect.min.pandemic}$ [ms]	4.0013	4.3653	1.4570	1.4647
$T_{infect.min.endemic}$ [ms]	40.0022	43.6389	14.5488	14.5628
$T_{infect.min.contagion}$ [ms]	40.0022	43.6389	14.5488	14.5628

Table 19 summarizes the minimum amount of hops required for infecting all critical nodes in the topology, which includes all nodes required to successfully conduct a destruction attack. Furthermore, it shows the resulting $T_{infect.min.}$ which is calculated with the sum of propagation and transmission duration per hop. For instance, the first three topologies have a central control node which is sufficient to shut down the power grid, thus, the attack does not need to infect all ML-nodes. This results in, e.g., two hops for the centralized topology, i.e., LL-node to ML-node to HL-node, which are added by their transmission- and propagation delays. The decentralized topology does not have a central control node which means that all ML-nodes must be infected to successfully attack the entire power grid.

$T_{infect.min.}$ could be calculated for all other attack types, e.g., if the attacker aims to spy on all ML-nodes, or the attacker aims to spy on all LL-nodes. However, we restrict all further metrics to our worst case scenario, i.e., the destruction attack, intentionally inducing a blackout. We detail on this in the improvement section 11.7. Therefore, we target either the HL-node or all ML-nodes.

6.5 Infection Efficiency

We define a time critical metric, namely the *infection efficiency* ($E_{infection}$), that represents the attack speed. Since we assume zero day vulnerabilities in all nodes and for all simulations, we do not consider differences in hardware or software, cf. Section 4.6. $E_{infection}$ represents the theoretical minimum reaction time available to defenders before all critical nodes are compromised. This point is reached either by infecting the control center ($T_{C.Center}$) or all gateway nodes ($T_{last.GW}$). At this point the attacker has control over all subjacent nodes. If all critical nodes are infected early in the simulation, this metric increases. $E_{infection}$ is defined by the minimum malware infection time ($T_{infect.min.}$) that is theoretically possible, divided by the measured infection time of the critical nodes ($T_{C.Center}$ or $T_{last.GW}$), whichever comes first:

$E_{infection} \in [0, 1]$ where 1 means that malware manages to infect the critical nodes in $T_{infect.min.}$ and 0 that not all critical nodes are infected or ∞.

$$E_{infection} = \begin{cases} \frac{T_{infect.min.}}{T_{C.Center}} & \text{for topologies with a control center} \\ \frac{T_{infect.min.}}{T_{last.GW}} & \text{for topologies without a control center} \end{cases} \tag{3}$$

6.6 Scanning Stealthiness

We characterize the scanning stealthiness of malware by its scanning behavior, i.e., how much "noise" a malware generates when scanning the network for new victims (using ARP-scanning, cf. Section 4.4.2). We define the *scan ratio* (R_{scn}) which represents the ratio of (successfully and unsuccessfully) scanned addresses to all theoretically possible scans.

Notation:
n_{host}(i) = Existing number of nodes in sub-network i
n_{addr}(i) = Number of theoretically available addresses per sub-network (i)
n_{scn}(i) = Number of all scans for sub-network i
R_{scn} $\in$ [0, 1] where 0 means no node, and 1 that all nodes are scanned

$$R_{scn} = \frac{\sum_{i=1}^{s} n_{scn}(i)}{\sum_{i=1}^{s} n_{host}(i) * (n_{addr}(i) - 1)} \tag{4}$$

The denominator in formula 4 defines the scanning space, i.e. the number of theoretically possible scans, when each node scans other nodes at most once but not itself:

$$\sum_{i=1}^{s} n_{host}(i) * (n_{addr}(i) - 1) \tag{5}$$

Therefore, each address may be scanned multiple times by different nodes if a malware is not sophisticated to coordinate the scanning behavior. Different scanning strategies discussed in Section 4.4.2, and defined for our malware models in Sections 4.5.1 through 4.5.3, dictate whether a source node scans the entire address space, a reduced address space, or does not scan at all.

Figure 23 shows an example of the scanning space for one subnet. Red dots represent a scan that has been performed from a specific host (y-axis) to an address of the address space (x-axis). Green dots represent the host did not scan the particular address. The diagonal line of green dots identifies nodes never scanning themselves. For our model we assume that one node scans one target at most once, thus, no re-scanning by the same node if a scan packet is lost. Using the scanning space, cf. Formula 5, the *scan ratio*, cf. Formula 4, is defined as the number of scans that were performed, divided by the theoretically possible scans. A higher value represents more detectable traffic, thus, a lower scanning stealthiness. The ratio of *unscanned addresses* (R_{uscn}) is illustrated by the green dots in Figure 23. It is defined as follows:

$$R_{uscn} = 1 - R_{scn} \tag{6}$$

We represent scanning "noise" with the indicators R_{scn} and R_{uscn} rather than illustrations. Therefore, Figure 23 is just shown as an example and will not be repeated for all malware simulations.

6.7 Scanning Efficiency

Furthermore, we define the *efficiency of scanning* (E_{scn}) that represents how efficient a malware type is in discovering new targets among the available addresses. As discussed in Section 3.3, ML-nodes and the HL-node have more than one network interface. However, we count them as one because the hitlist, that is generated after infection, includes all interfaces of this particular node. We define limiting boundaries for Formula 7, namely that each node may only use one interface per network. Therefore, we calculate this metric with infected nodes rather than interfaces. A higher value represents a less "noisy" scanning strategy. E_{scn} defines the infection ratio of those nodes that are infected by scanning in the network to the number of total scans. It is defined as follows:

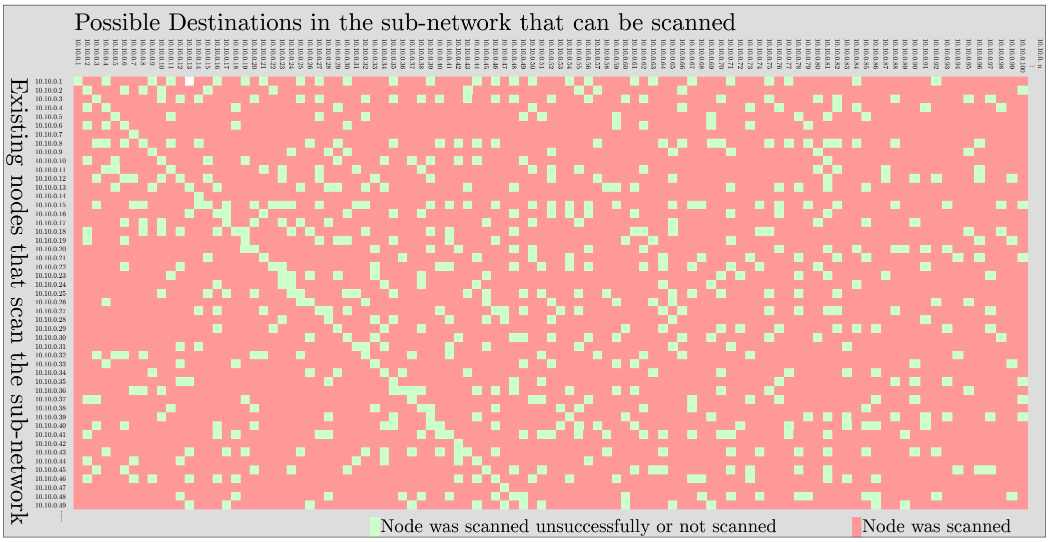


Figure 23: Scan ratio of pandemic malware

$E_{scn} \in [0, 1]$

$$E_{scn} = \frac{(\sum_{i=1}^{s} n_{inf}(i)) - 1}{\sum_{i=1}^{s} n_{scn}(i)} \tag{7}$$

where

$$\left(\sum_{i=1}^{s} n_{inf}(i)\right) - 1 \tag{8}$$

represents all nodes that are scanned across all sub-networks *excluding patient-zero* which is infected manually, thus, not scanned. Therefore, the maximum efficiency is 100% when each scan results in one infection.

E_{scn} expresses a very strict definition of efficiency. 100% scanning efficiency can only be achieved in two cases:

- One source node per sub-network scans all existing nodes and every scan is a success. No packet loss occurs and none of the other hosts participate in scanning. In this case all the scanning effort needs to be taken over by one node, decreasing the scanning speed. In addition, the scanning source is easily detected if the network is observed.
- The scanning is highly coordinated. One possibility is to use sequential scanning where each node only scans and infects one node, then stops, and the following node continues. Sequential scanning is slow and fails if a scanning packet gets lost. An alternative is to use some control traffic (C&C) to coordinate the scanning. But such control traffic requires additional effort and also reduces the stealthiness.

In the optimal case the number of scans is equal to the number of existing nodes excluding patient zero, i.e., $E_{scn} = 100\%$. However, if more than one node scans the same target on a sub-network or nodes scan unassigned addresses, the efficiency drops considerably. In reality, attackers aim to optimize the scanning strategy depending on the sophistication level of their malware. Decreasing scanning output to reach 100% efficiency would require perfect coordination, thus shifting this effort to control traffic, including detailed knowledge of the topology. Since we assume self-propagating malware in this work, we do not simulate malware with sophisticated C&C structures.

This metric is not applicable for malware types that are not scanning other nodes, e.g., contagion malware. Then E_{scn} is not calculated as 100%, because it would represent a division by 0, since no scanning occurs. Therefore, we use the notation "n.a." for these cases.

6.8 Propagation Stealthiness

In addition to scanning for targets, malware needs to propagate itself, i.e. sending the actual payload to the victims. This is another activity detectable in the network that reduces stealthiness. We define the propagation behavior of malware that utilizes *unsolicited traffic* (U_{tr}) as measure for the visibility of payload propagation. U_{tr} describes what percentage of traffic [Bytes] in a certain time frame has illegitimate origin, thus, seems suspicious because it is not invoked by legitimate smart grid applications. U_{tr} represents the visibility of unsolicited payload propagation. It is defined as follows:

Notation:
T_{active} = Time of malware activity to normalize infectious- with overall traffic such that: $\mathrm{B}_{\mathrm{unsol.}} = 0$ for all $\mathrm{t} > \mathrm{T}_{\mathrm{active}}$
$B_{unsol.}(T_{active}, i)$ = Bytes associated with unsolicited traffic in sub-network i during interval $[0, \mathrm{T}_{\mathrm{active}}]$
$B_{total}(T_{active}, i)$ = Bytes of total traffic in sub-network i during interval $[0, \mathrm{T}_{\mathrm{active}}]$
U_{tr} $\in [0, 1]$ where 0 represents no suspicious traffic

$$U_{tr} = \frac{\sum_{i=1}^{s} B_{unsol.}(T_{active}, i)}{\sum_{i=1}^{s} B_{total}(T_{active}, i)} \tag{9}$$

The time of *malware activity* (T_{active}) needs to be defined, because the activity time differs for different malware types. For comparable results we always compare the unsolicited traffic during the active time interval with the total traffic in the same active time interval.

Additionally, we define another metric also representing the stealthiness of malware based on suspicious connections from the expected origin of *network* *flows*, in our case TCP flows. We name it *unsolicited flow* (U_{flow}). It is defined by the number of unsolicited flows in comparison to all flows within the time interval [0, T_{active}], thus, it represents the visibility of unsolicited network flows. Although this metric does not convey information about the amount of data transported, it does represent the amount of connections established. These connections could be easily detected by an IDS.

Notation:
$F_{unsol.}(T_{active},i)$ = Number of unsolicited flows in sub-network i during time interval [0, $\mathrm{T}_{\mathrm{active}}$]
$F_{total}(T_{active},i)$ = Number of flows in overall active traffic in sub-network i during time interval [0, $\mathrm{T}_{\mathrm{active}}$]
U_{flow} $\in$ [0, 1] where 0 represents no unsolicited flows

$$U_{flow} = \frac{\sum_{i=1}^{s} F_{unsol.}(T_{active}, i)}{\sum_{i=1}^{s} F_{total}(T_{active}, i)} \tag{10}$$

Both metrics, U_{tr} and U_{flow}, require the network operator to implement appropriate monitoring solutions that can interpret network traffic patterns.

6.9 Anomaly Detection specific to Contagion Malware

Since the contagion malware is sending its payload in existing flows, the same methods used for pandemic and endemic malware are unable to detect any network anomalies. Therefore, we introduce a more specific approach, namely *anomalous flow detection* (A_{flow}). We aim to detect patterns diverging from expected behavior in network traffic that requires the correlation of, packet size and direction within existing legitimate flows, in our case TCP flows. This distinct anomaly output of injecting the payload at the end of the communication flow is applicable to contagion malware in our model. Furthermore, this anomalous/malicious payload can target victims in either direction, from source to destination, or destination to source. Therefore, infected hosts can infect victims by either sending data, e.g., from LL-node to ML-node to HL-node (upstream), or by reverse injecting the malicious payload to inbound flows, e.g., HL-node to ML-nodes to LL-nodes (downstream). However, the legitimate data is only sent upstream from source to destination in both cases. Furthermore, we assume that network operators can predict what flow behavior can be expected, due to the nature of M2M communication, and what it should look like. This was outlined in Section 4.5. Therefore, A_{flow} represents the visibility of hidden payload propagation. Therefore, contagion malware do not open their own unsolicited connections, thus, transfer the payload covertly inside legitimate flows.

Notation:

$F_{covert}(T_{active}, i)$ = Number of flows in sub-network i that contain malicious data during time interval $[0, \mathrm{T}_{\mathrm{active}}]$

A_{flow} $\in [0, 1]$

$$A_{flow} = \frac{\sum_{i=1}^{s} F_{covert}(T_{active}, i)}{\sum_{i=1}^{s} F_{total}(T_{active}, i)} \tag{11}$$

Figure 24 illustrates a TCP flow from LL-node to ML-node, which seems, for the most part unsuspicious. However, at the end anomalies occur, when the malicious payload shows a significantly increased file size compared to legitimate data. This could be detected by IDS that can anticipate the expected traffic patterns. Should the malicious payload be better obfuscated among native traffic, thus, not stand out by it size, then only deep packet inspection methods can present a chance to detect malicious behavior.

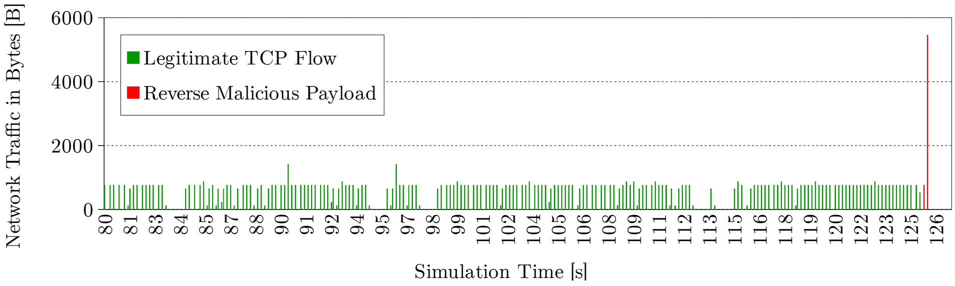


Figure 24: Covert communication within legitimate data flows

6.10 Propagation Efficiency

We define the metric *propagation efficiency* ($E_{propag.}$) to consolidate the inverse of the propagation stealthiness metrics. It represents the visibility of malware payloads traveling the network, excluding scanning traffic, cf. Sections 4.4.3 and 6.8. It is based on the metrics U_{TCP} (unsolicited flow) and A_{flow} (covert communication) both of which exclude each other in our model, thus, only one can have a positive value while the other is zero. We calculate the propagation efficiency as follows:

$E_{propag.} \in [0, 1]$

$$E_{propag.} = \begin{cases} 1 - U_{flow} \text{ for unsolicited connections} \\ 1 - A_{flow} \text{ for covert communication} \end{cases} \tag{12}$$

6.11 Noise-Suppression Efficiency

We define the *noise-suppression efficiency* of malware ($E_{noise.suppress.}$) as a metric that consolidates all detectable signals in the network cf. Formula 13.

- First, the *scanning efficiency* (E_{scn}), as introduced in Section 6.7, represents the target discovery process. Therefore, high scanning efficiency produces less detectable noise in the network. Although this works for pandemic and endemic malware, the contagion malware does not scan the network. Therefore, we weigh its scanning traffic with 0 for the contagion case, cf. Table 20.
- Next, the *propagation efficiency* ($E_{propag.}$), as introduced in Section 6.10, represents the payload propagation in the network. High propagation efficiency represents decreased network detectability.
- Finally, we include C&C detection, as discussed in the generic attack life-cycle model, cf. Section 4.4. However, we do not simulate C&C traffic in our model, thus, it is weighed with 0 in Table 20. However, we do include it for completeness and reusability in future works.

Notation:
w_{scn} = Weight for scanning traffic
$w_{propag.}$ = Weight for propagation traffic
$E_{C\&C}$ = C&C efficiency (not defined in detail)
$w_{C\&C}$ = Weight for C&C traffic
$E_{noise.suppress} \in [0, 1]$ where 1 represents perfect noise suppression

$$E_{noise.suppress.} = w_{scn} * E_{scn} + w_{propag.} * E_{propag.} + w_{C\&C} * E_{C\&C} \tag{13}$$

Whereas,

$$w_{scn} + w_{propag.} + w_{C\&C} = 1 \tag{14}$$

We define the formula in such a way that each part can be weighed individually. However, we include Table 20 to outline the settings for our simulation model. These could be adapted for different simulations. Since each malware model has special features, e.g., pandamic malware scans noisily but has a small payload, whereas endemic malware scans more quietly but has a large payload, we expect to see very different behavior for this metric.

Table 20: Weights for different detectable network signals

Malware Type	$\mathrm{w}_{\mathrm{scn}}$	$\mathrm{w}_{\mathrm{propag.}}$	$\mathrm{w}_{\mathrm{C\&C}}$[1]
Pandemic	0.5	0.5	0
Endemic	0.5	0.5	0
Contagion	0	1	0

Notation: (1) C&C is not simulated in our model

6.12 Malware Attack-Efficiency

We define the *attack-efficiency* of malware (E_{attack}) as a metric that consolidates noise-suppression, cf. Section 6.11, and infection efficiency, cf. Section 6.5. Although this metric can be calculated for all attack types, e.g., disruption attacks, data theft attacks, among others, we restricted it to the scenario of destruction attacks, i.e., our worst case scenario.

Notation:
$w_{infection}$ = Weight for infection efficiency
$w_{noise.suppress.}$ = Weight for noise suppression efficiency
E_{attack} $\in$ [0, 1] where 1 represents perfect attack efficiency

$$E_{attack} = w_{infection} * E_{infection} + w_{noise.suppress.} * E_{noise.suppress.} \tag{15}$$

Whereas,

$$w_{infection} + w_{noise.suppress.} = 1 \tag{16}$$

We define the formula in such a way that each part can be weighed individually. However, we define that the distribution for our model is equal, thus, $w_{infection}$ and $w_{noise.suppress.}$ are each 0.5.

6.13 Attack-Defendability

The *attack-defendability* (D_{attack}) is defined as the reverse of E_{attack}. It describes how well defenders can detect an ongoing attack for initiating countermeasures.

$$D_{attack} = 1 - E_{attack} \tag{17}$$

6.14 Theoretical extensions for our simulation model

Up to this point we introduced calculable metrics that represent the performance of different malware types. However, we are leaving the scope of our simulation model because we also consider some defensive measures, cf. Section 10, which are not simulated in our model. To clarify, we illustrate which metrics affect each other, whereas E_{attack} and D_{attack} yield calculable results, thus, located in the experimental half of Figure 25. However, other metrics introduced below, i.e., C_{attack} and R_{attack} do not yield calculable results. The proactive and reactive measures, cf. Section 10, cover all parts of the malware attack life-cycle model, thus, exceed the scope of our simulation model. Therefore, we cannot quantify these defensive measures, and their corresponding metrics in our simulations.

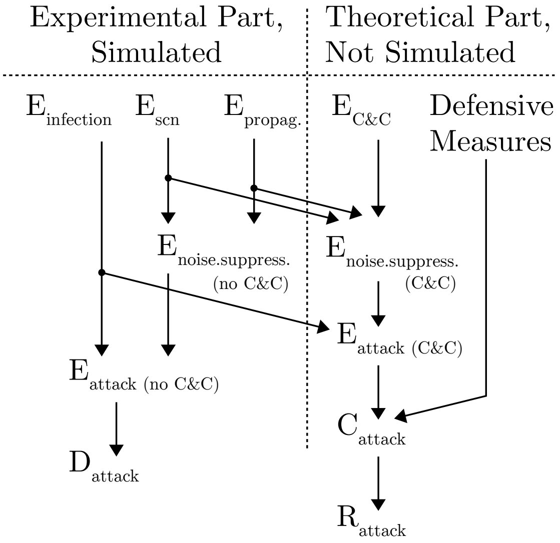


Figure 25: Differentiation of experimental and theoretical analysis

6.15 Attack-Containment

We define *attack-containment* (C_{attack}) to accommodate future works with suitable metrics for each defense measure. The containment measures represent the counterparts of attack-efficiency and noise-suppression efficiency, in Formulas 13 and 15, thus, attack-containment reduces the attack-efficiency. Additionally to *infection containment* ($C_{infection}$), *scanning containment* (C_{scn}), and *propagation containment* ($C_{propag.}$), we include *C&C traffic containment* ($C_{C\&C}$) to provide a complete set of metrics.

Notation:
C_{attack} = Sum of containment measures that reduce attack-efficiency
$C_{infection}$ = Sum of measures that reduce the infection efficiency
C_{scn} = Sum of measures that reduce the scanning efficiency
$C_{propag.}$ = Sum of measures that reduce the propagation efficiency
$C_{C\&C}$ = Sum of measures that reduce the C&C efficiency
$C_{attack} \in [0, 1]$

$$\begin{aligned} C_{attack} = & w_{infection} * C_{infection} + w_{noise.suppress.} * \\ & (w_{scn} * C_{scn} + w_{propag.} * C_{propag.} + w_{C\&C} * C_{C\&C}) \end{aligned} \tag{18}$$

To clarify how the sum of each containment measure is constructed, we discuss a *generic example* ($C_{example}$) to attribute a mix of proactive and reactive measures to each of them.

Notation:
M_{pro} = Applicable proactive defense measure, cf. Section 10
M_{re} = Applicable reactive defense measure, cf. Section 10
w_{pro} = Weight of proactive measures
w_{re} = Weight of reactive measures
p = Proactive measure number p
r = Reactive measure number r
$C_{example} \in [0, 1]$

$$C_{example} = \sum_{i=1}^{p} \sum_{j=1}^{r} w_{pro}(p) * M_{pro}(p) + w_{re}(r) * M_{re}(r) \tag{19}$$

Therefore, $C_{example}$ should represent the sum of all effective proactive and reactive defensive measures that mitigate its counterpart $E_{example}$. For instance, $C_{propag.}$ includes measures that can effectively counteract and mitigate the propagation behavior of malware in a network. Therefore, it includes (p=2) proactive measures, e.g., whitelisting on firewalls and controlled user access, and (r=2) reactive measures, e.g., anomaly detection and anti-malware tools. This example can be expanded upon additional measures.

6.16 Attack-Resilience

We define *attack-resilience* (R_{attack}) as a metric that represents the ability of a system to withstand cyber-attacks. R_{attack} should be calculated by starting from 1 (secure against attacks), which is reduced by the attack efficiency, which is then counteracted by the containment measures. We define R_{attack} as follows:

$R_{attack} \in [0,\ 1]$

$$R_{attack} = 1 - E_{attack} + C_{attack} \tag{20}$$

As discussed in Section 1.7, the term "attack-resilience" is introduced in analogy to the term "resilience" in the context of equipment failure. The term "resilience" is broadly used. However, it is reserved for the resistance of equipment or a system to fail. Such failure need not necessarily be connected to an attack. However, our approach defines how well a system is equipped against artificially induced failures, e.g., deliberate shutdowns [14, 160].

6.17 Summary of all Notations

Table 21 summarizes all notations introduced in the metrics.

Table 21: Summary of all notations in Section 6

Notation	**Explanation**
s	Number of sub-networks
n_{inf}(i)	Number of infected nodes in sub-network i
n_{host}(i)	Existing number of nodes in sub-network i
R_{inf}	Infection ratio
R_{clean}	Ratio of clean nodes
$T_{first.GW}$	Time until first gateway is infected
$T_{C.Center}$	Time until control center is infected
$T_{last.GW}$	Time until last gateway is infected
$T_{75\%.nodes}$	Time until 75% of all nodes are infected
$T_{last.node}$	Time until the last field node is infected. $T_{last.node} \leq T_{all.nodes}$
$T_{all.nodes}$	Time until all nodes on the network are infected. May be ∞
$T_{infect.min.pandemic}$	Theoretical minimum infection duration for pandemic malware
$T_{infect.min.endemic\_contagion}$	Theoretical minimum infection duration for endemic and contagion malware
$E_{infection}$	Infection efficiency
n_{host}(i)	Existing number of nodes per sub-network i
n_{addr}(i)	Number of theoretically available address per sub-network i
n_{scn}(i)	Number of all scans per sub-networks i, cf. Figure 23
R_{scn}	Scan ratio
R_{uscn}	Ratio of unscanned nodes
E_{scn}	Scanning efficiency
T_{active}	Time of malware activity, to normalize infectious traffic to overall traffic
$B_{unsol.}(T_{active}, i)$	Bytes associated with unsolicited (malicious) traffic in subnet i during interval $(0, T_{active})$
$B_{total}(T_{active}, i)$	Bytes of effective overall traffic (legitimate and malicious) in subnet i during $(0, T_{active})$
U_{tr}	Unsolicited traffic in byte
$F_{unsol.}(T_{active}, i)$	Number of TCP flows associated with unsolicited traffic in subnet i during interval $(0, T_{active})$
$F_{total}(T_{active}, i)$	Number of TCP flows in effective overall traffic in subnet i during interval $(0, T_{active})$
U_{flow}	Unsolicited traffic flow, in our case TCP
A_{flow}	Anomalous flow detection

... Continued on next page

Notation	**Explanation**
$E_{propag.}$	Propagation efficiency
w_{scn}	Weight for scanning traffic
$w_{propag.}$	Weight for propagation traffic
$E_{C\&C}$	C&C efficiency
$w_{C\&C}$	Weight for C&C traffic
$E_{noise.suppress.}$	Noise.suppression Efficiency
$w_{infection}$	Weight for infection efficiency
$E_{noise.suppress.}$	Weight for noise suppression efficiency
E_{attack}	Attack efficiency
D_{attack}	Attack defendability
${C_{attack}}^{t}$	Sum of measures that reduce the attack-efficiency
${C_{infection}}^{t}$	Sum of measures that reduce the infection efficiency
${C_{scn}}^{t}$	Sum of measures that reduce the scanning efficiency
${C_{propag.}}^{t}$	Sum of measures that reduce the propagation efficiency
${C_{C\&C}}^{t}$	Sum of measures that reduce the C&C efficiency
${M_{pro}}^{t}$	Applicable proactive defense measure
${M_{re}}^{t}$	Applicable reactive defense measure
${w_{pro}}^{t}$	Weight of proactive measure
${w_{re}}^{t}$	Weight of reactive measure
p^{t}	Proactive measure number p
r^{t}	Reactive measure number r
${R_{attack}}^{t}$	Attack resilience

Notation: (t) Theoretical metric

Concluded

7 Architecture Results and Discussion

Notice of adoption from previous publications:
Parts of this chapter have been previously published in [44], including the results from our topological analysis. The main contribution of the author was the development of evaluation metrics. The co-authors contributed valuable input on additional metrics, and their validation. The results were also presented in [37, 39].

According to research question 1, cf. Section 1.3, we elaborate which communication topology delivers the most promising features in terms of security by design.

7.1 Network Topologies for Critical Infrastructures

We provide a qualitative comparison of the ICT topologies introduced in Section 3.4. Table 22 shows their applicability for the proposed reference architectures elaborated in Section 2.2. While centralized and cell topologies can be supported by most existing reference architectures, mesh and decentralized topologies are not compatible with all. The ENSIA model emerges to be able to support the widest range of topologies. The quality indicators introduced in Section 3.4 are discussed and compared in Table 23.

Table 22: Mapping topologies to reference architectures, as presented in [44]

Topologies, cf. Figure 7	SGAM [21]	BSI [61]	ILO [78]	NIST [174]	ENISA [49]
a. Centralized	✓	✓	–	✓	✓
b. Cell design	✓	✓	✓	✓	✓
c. Mesh design	–	–	~	–	✓
d. Decentralized	–	–	–	–	~

Notation: (✓) Yes, (~) Option, (–) No

The following statements summarize the most prominent results concerning the network architecture:

- The main drawback of fully *centralized topologies*, cf. Section 3.4.1 and Figure 7-a, is the potential for communication bottlenecks between LL-ICT and HL-ICT. Furthermore, central control systems are an easy target for attackers, even if redundancies are implemented, making well established defense measures all the more important. However, centralized legacy SCADA systems may persist as they exist today, considering they are well tested, established, and understood.

- In the *cell topology*, cf. Section 3.4.2 and Figure 7-b, cells can operate autonomously. A local cell controller (ML-node) manages energy consumption and communication as a local master node. Strict policies can prevent malware from spreading vertically upgrading the cell controller to an intermediary firewall unit. Horizontal propagation can be contained by limiting communication among cells. However, each cell represents a small hierarchical structure subjacent to the ML-ICT as the convergence entity, making the cell controller the most vulnerable point. Several cell controllers are connected via a mesh network for added resilience against failures.
- The *mesh topology*, cf. Section 3.4.3 and Figure 7-c, connects all devices at the low and medium ICT level into mesh networks. This approach provides better resilience against equipment failure. However, it comes at the cost of decreased containment capabilities because these mesh networks provide many alternative distribution paths. Furthermore, HL-nodes (SCADA systems) cannot be integrated directly except via dedicated gateways. Although they are similar to the previously discussed cell controllers, these gateways do not restrict the formation of a mesh network across separate grid types, e.g., power grid and water grid, or other power grids from neighboring districts. Even water or gas grids could participate in such a mesh network.
- A fully *decentralized topology*, cf. Section 3.4.4 and Figure 7-d, reveals its main drawback in malware containment, however, at the benefit of increased resilience against failure. Malware may quickly spread to other grid types and across hierarchies. Moreover, today's power transmission grids are controlled exclusively by centralized SCADA systems. Upgrading all of these existing systems for mesh network capability is likely not economically viable.

Table 23 provides a summary of the quality indicators in the ICT topologies across all hierarchy levels, providing an estimation of the impact on different levels of the hierarchy. Table 24 compares the quality indicators in detail for the four ICT topologies. It shows the benefits and drawbacks of each topology as discussed in Sections 3.4.1 through 3.4.4.

Based on the topology results we expect to see the slowest propagation of malware via mesh networks, because the wireless communications link provides slower connections than does the optical fiber link. However, the PLC link is even slower in terms of bandwidth, thus, it will represent the bottleneck for the centralized and cell topology. Mesh networks provide the highest connectivity giving attackers many paths to discover other nodes on the network. This leads to decreased security against horizontal and vertical infections.

Table 23: Comparing topologies over hierarchy levels, as discussed in Section 3

		ICT Hierarchy Level			
		HL-ICT	ML-ICT	LL-ICT / Building	
Centralized	Resource control	✓	✓	✓	✓
	Security	✓	✓	✓	✓
	Resilience	–	–	–	–
	Quality of service	–	–	–	–
	Compatibility	✓	✓	✓	✓
	Cost	✓	∼	–	–
Cell Topology	Resource control	✓	∼	✓	✓
	Security	✓	∼	✓	✓
	Resilience	–	✓	–	–
	Quality of service	–	✓	–	–
	Compatibility	✓	∼	∼	∼
	Cost	–	✓	–	–
Mesh Topology	Resource control	✓	∼	–	–
	Security	✓	∼	–	–
	Resilience	–	✓	✓	✓
	Quality of service	–	✓	✓	✓
	Compatibility	✓	∼	–	–
	Cost	–	✓	∼	∼
Decentralized	Resource control	∼	∼	–	–
	Security	–	–	–	–
	Resilience	✓	✓	✓	✓
	Quality of service	✓	✓	✓	✓
	Compatibility	–	–	–	–
	Cost	✓	∼	∼	∼

Notation: (✓) High or Strong, (∼) Medium, (–) Low or Weak

Table 24: Comparison of quality indicators, as presented in [44]

	Centralized	Cell design	Mesh design	Decentralized
Resource control:				
Situational awareness for top level decisions	✓✓	✓	-	- -
Situational awareness for resource optimization	✓✓	✓	-	- -
ICT topology matches electrical topology	✓	✓✓	-	-
Topology re-negotiation after failure	✓✓	-	-	-
Complex data collection	-	✓✓	✓	-
Security:				
Horizontal security against malware propagation	✓✓	✓✓	- -	- -
Vertical security against malware propagation	✓	✓	-	- -
Physical security against direct access	✓	✓	- -	- -
Data security through local data management	✓	✓	-	-
Resilience:				
Resilience against failure	✓	✓✓	✓✓	✓✓
Resilience against attacks	- -	✓	✓	✓
Local intelligence	- -	✓✓	✓✓	✓✓
ICT and electrical cells form micro-grids	- -	✓✓	-	-
Complexity of control over nodes	✓	-	✓	✓
Self-organization	-	✓	✓	✓✓
Quality of service:				
Low latency	- -	✓✓	✓✓	✓✓
Low congestion	- -	✓	✓✓	✓✓
Compatibility:				
SCADA can be integrated into the topology	✓✓	✓	-	- -
Cost:				
Low cost upgrade feasible	✓	-	-	- -

Notation: (✓) Benefit, (-) Drawback

8 Malware Results and Discussion

Notice of adoption from previous publications:
Parts of this chapter have been previously published in [48], investigating malware capabilities and analyzing their distinct features, future developments and smart-grid-specific attack vectors. These malware capabilities represent our malware models from Section 4. The main contribution of the author was the extraction of all features, use-cases from todays Internet, and the development of future attack vectors. The co-authors contributed valuable input on additional metrics, discussions, and feature selection. Also presented in [38, 40, 45].

We include an in depth classification of existing malware, the future evolution of malware development, and attack types that are specificity tailored to smart grids.

8.1 Classification of Existing Malware

In this section we evaluate 19 representative malware types using metrics that correspond to the modules in our generic attack-life-cycle model introduced in Section 4.4. Our samples include well-researched malware types from the Internet that represent a wide range of features relevant to smart grid attacks. Some types included in the sample are older but prevailing malware at various levels of sophistication. Others are modern, highly evolved APTs-produced malware. We aim to provide insight into recent developments and extrapolate future threats, but are aware that there are many more samples available for analysis. Therefore, we have selected examples that either represent the most important classes, or showing variants of modern techniques, or their most recent representatives. Since the primary difference between smart grids and the Internet is the former's predominant use of, e.g., M2M communication as well as the homogeneity of its devices, cf. Section 2.2.7, this selection takes into account lateral attack vectors via the enterprise network into the control network. The large number of possible attacks allows us to infer similar behavior to that of Internet malware [48].

All metrics are processed in Tables 25 through 28 and sorted chronologically by year of detection. We provide an overview of the metrics below and reference the corresponding sections within the generic model [48]:

- General information (Table 25)
  - Access vectors, cf. Section 4.4.1
  - Attack goals, cf. Section 4.4.6
  - References for all malware types

- Propagation (Table 27)
    - Propagation vectors, cf. Section 4.4.3
    - Payload types, cf. Section 4.4.3
- Communication patterns (Table 27)
    - Discovery methods, cf. Section 4.4.2
    - Network protocols, cf. Section 4.4.3
    - C&C methods, cf. Section 4.4.5
- Persistence and defense-mechanisms (Table 28)
    - Morphism level, cf. Section 4.4.4
    - Evasion of anti-malware tools, cf. Section 4.4.1
    - Memory residency, cf. Section 4.4.1
    - Code obfuscation, cf. Section 4.4.1
    - Service injection, cf. Section 4.4.1
    - Uninstall mechanisms, cf. Section 4.4.1
    - Rapid reinfection, cf. Section 4.4.1
    - Cleanup mechanisms, cf. Section 4.4.1

We further supplement the aforementioned metrics with the following four qualitative metrics: [48]

- Level of complexity, cf. Table 25
- Propagation speed, cf. Table 27
- Propagation scope, cf. Table 27
- Modularity, cf. Table 27

In this comparison we generally disregard infection methods and thus the known vulnerabilities, as we could only speculate which vulnerabilities will emerge in future. The present examples show that only a few malware instances exploit the same vulnerabilities, which means they have access to a large set of possibilities. But exceptions can still be found in which new malware recycles known vulnerabilities or members of the same family take advantage of the same vulnerabilities, which is the case for most Stuxnet derivatives. Since we consider only well-researched malware types, the vulnerabilities used by known malware can today be counteracted by keeping security implementations up-to-date. However, new vulnerabilities arise daily, and malware authors are becoming increasingly creative [48].

8.1.1 General Information

Section 4.4.1 and Table 25 describe detailed examples of access methods. Furthermore, the chronological evolution is a strong indication that recent malware development is shifting away from disruption attacks and rather focusing on data theft. However, it is worth noting that this evolution does *not* mean that DDoS attacks are no longer feasible or not of interest. The modular concept of today's malware allows for easy addition of new modules at any time. This means that DDoS capability could be added as part of an upgrade to existing malware [48].

Based on our investigation, we argue that future malware may increasingly have the capability of installing modules with the goal of physical destruction. Attacks that target cyber-physical systems, e.g., Stuxnet or BlackEnergy [56,123] utilize extensive data theft capabilities along with a destructive payload. This helps the attacker to map the environment, as it has to bypass the defenses of industrial networks. Scanning and credential theft are among the methods utilized. Technically, most sophisticated data theft malware only lack modules for targeting cyber-physical equipment as well as protocols in order to become a threat to smart grids. This means that utilities are well advised to defend against data-theft-class malware, i.e. endemic malware and above, based on the suspicion that cyber-physical attack capabilities may be implemented in the future [48].

Extortion attacks have been highly present in recent news and are feasible in smart grid attacks, as critical infrastructures provide strong leverage against communities. Extortion, via e.g. DDoS, could therefore precede destruction-attacks, as a modular extension. Furthermore, modern attacks are increasing in complexity, making them more challenging to defend against. Some examples show that the development of low-complexity malware is still conducted for monetary gain [9, 72, 89, 162]. Although, they are not to be underestimated, development is generally moving in the direction of universal capabilities [48].

Table 25: General malware information, as presented in [48]

Name	Year of Detection	Level of Complexity	Access Methods	Attack Goal	Reference
CodeRed1	2001	Low	S2S	Disruption	[23, 131, 166]
CodeRed2	2001	Low	S2S	Disruption	[23, 131, 147, 166]
Nimda	2001	Low	Email, C2S, S2C, H2NS	Disruption	[23, 166]
Slammer	2003	Low	C2S, S2C	Disruption	[114, 147]
Sality	2003	Medium	H2NS, RD, P2P	Data theft	[31, 175]
Conficker	2008	Low	C2C, P2P, RD	Data theft	[31, 59, 158]
Regin	2008	High	P2P, C2C	Data theft	[94, 169]
Aurora	2010	Medium	Email, S2C	Data theft	[95, 99, 123]
Stuxnet	2010	High	RD, C2C, P2P	Destruction, Data theft	[58, 99, 123]
Duqu	2011	High	Email, C2C, P2P, H2NS	Data theft	[13, 96, 99, 173]
Flame	2012	High	C2C, RD	Repurpose, Data theft	[13, 97, 99]
Gauss	2012	High	RD	Data theft	[13, 98, 99]
Duqu2	2014	High	Email, C2C, S2C	Data theft	[13, 96, 99]
Equation	2014	High	C2C, RD, S2C	Data theft	[95, 99]
BlackEnergy3	2015	High	RD, Email, C2C	Destruction, Data theft	[56, 99]
Cozy Duke	2015	High	Email, S2C, H2NS	Data theft	[57, 99]
AdWind	2015	High	Email	Data theft	[72, 89, 99]
Locky	2016	Low	Email, H2NS	Extortion, Disruption	[9, 162]
PLC-Blaster	2016	Low	C2C	Destruction, Repurpose	[108, 165]

Notation: (P2P) Peer-to-Peer, (S2S) Server-to-Server, (S2C) Server-to-Client, (C2S) Client-to-Server, (C2C) Client-to-Client, (H2NS) Host-to-Network-Share, (RD) Removable Drive (non-network), (Email) Phishing emails through active dispatch or passive third party (several protocols).

Based on Table 25, the following findings can be seen: [48]

- Overall malware complexity is increasing substantially.
- Attack goals are shifting increasingly toward more complex data theft, and away from disruption (DDoS).
- Disruption attacks are still feasible as part of a modular upgrade in more complex attack environments or low-complexity efforts.
- Destruction attacks are expected to become more common in the near future, when malware complexity increases.
- Sophisticated data theft malware currently lacks modules for cyber-physical attack capability and specific smart grid protocols to be deployed in smart grids. However, such features may be implemented in the future.
- Development is generally moving toward enabling more universal capabilities. Modularity enables fast upgrades.
- Access vectors are becoming more sophisticated and often depend on the availability of zero-day exploits.

8.1.2 Propagation

Table 26 reviews malware network activity in terms of scope, speed, propagation vectors, payload construction, and modularity. The propagation scope, which in some cases is global, is shifting increasingly toward targeted attacks. The scope could be sub-categorized and refined in more detail but we omit any finer granularity as to targeted persons, companies, or states. The Black-Energy and Regin attacks [56,94,113] demonstrated that all involved targets were spied upon regardless of their level of protection. Attacks against smart grid operators will almost certainly be of a targeted nature, as utilities are expected to implement strong defenses. Yet, there remains the possibility that non-targeted malware finds its way into a utility companies' network as occurred in the nuclear power plant in Gundremmingen, Germany [141]. This incident, however, did not result in a cyber-physical attack thanks to well-implemented network segmentation [48].

Concerning the propagation speed, development is converging toward stealthy malware types. These accept penalties in propagation speed in order to improve their stealthiness and persistence. Equation and Gauss [95, 98] in particular stand out for their very slow spreading speed in support of sophisticated stealthiness. Then, on the other hand, CodeRed and Nimda [23, 131, 147, 166] propagate at incredible speeds, producing many network anomalies that make them easily detectable. In between, however, there are many moderate types of malware that prioritize stealthiness in the interest of prolonged persistence [48].

The propagation vectors of malware are a key feature in cyber-attacks; here impeding metrics, e.g., scope and speed, play a crucial role. Propagation vectors correspond to the access methods in Table 26 and affect the movement of malware. Some only allow passive propagation via phishing emails, cf. AdWind and Locky [89, 162], or via removable drives, cf. Gauss and Equation [95, 98]. Many other types are capable of network-enabled self-propagation, which opens up many opportunities for rapid malware spread, e.g., Conficker [158]. The latter is capable of infecting other hosts via remote execution of a dropper file on networks using P2P networks, whereas Stuxnet [123] and Flame [97] partially utilize the same vulnerabilities, yet are more advanced in network propagation. They can also propagate via removable drives that allow access into air-gapped networks [48].

All propagation vectors are promising candidates for networks with homogeneous equipment types, as are expected in smart grids. Flame [97], for example, is capable of disguising itself as an update server while infecting hosts. Such a vector is essentially feasible with every update service. Given the fact that modern malware most often has a second-channel payload type, the increasing modularity fits in well as can be seen in Table 26. The analysis reveals that self-carried malware is rarely modular, whereas second-channel malware is predominantly modular. The payload types are discussed in detail in Section 4.4.3. Although there are still some self-carried payload types the trend is clearly moving toward second-channel payload propagation [48].

This comparison underlines that modern highly complex and APTs-made malware is increasingly modular and extensible. The number of functions seems to increase with every generation. Some modern malware types already have about 100 modules as exemplified by Duqu2 [96], whereas older malware types exhibit less flexibility. There are some exceptions, most notably PLC-Blaster [165], which is in an early stage of development yet highly specialized toward, in this case, industrial control equipment. If such capabilities are incorporated into a feature-rich modular malware comparable to [96], power utilities will face highly challenging adversaries [48].

Table 26: Malware propagation, as presented in [48]

Name	Scope	Speed	Vectors	Payload	Modularity
CodeRed1	Global	Aggressive	S2S	Self-carried	None
CodeRed2	Global	Aggressive	S2S	Self-carried	None
Nimda	Global	Aggressive	Email, C2S, S2C, H2NS	Self-carried	None
Slammer	Global	Aggressive	C2S, S2C	Self-carried	None
Sality	Global	Aggressive	H2NS, RD, P2P	Second-channel	Low
Conficker	Global	Aggressive	C2C, P2P, RD	Self-carried	Low
Regin	Targeted	Moderate	P2P, C2C	Second-channel	Medium
Aurora	Targeted	Aggressive	Email, S2C	Second-channel	Low
Stuxnet	Targeted	Moderate	RD, C2C, P2P	Second-channel	Medium
Duqu	Targeted	Moderate	Email, C2C, P2P, H2NS	Second-channel	Medium
Flame	Targeted	Moderate	C2C, RD	Second-channel	High
Gauss	Targeted	Passive	RD	Embedded	High
Duqu2	Targeted	Moderate	Email, C2C, S2C	Second-channel	High
Equation	Targeted	Passive	C2C, RD, S2C	Embedded	High
BlackEnergy3	Targeted	Moderate	RD, Email, C2C	Second-channel	Medium
Cozy Duke	Targeted	Moderate	Email, S2C, H2NS	Second-channel	Medium
AdWind	Targeted	Passive	Email	Second-channel	High
Locky	Global	Passive	Email, H2NS	Second-channel	Low
PLC-Blaster	Targeted	Moderate	C2C	Self-carried	None

Notation: (P2P) Peer-to-Peer, (S2S) Server-to-Server, (S2C) Server-to-Client, (C2S) Client-to-Server, (C2C) Client-to-Client, (H2NS) Host-to-Network Share, (RD) Removable Drive (non-network), (Email) Phishing through active sending or passive third party (several protocols).

Based on Table 26, the following findings can be seen: [48]

- Propagation scope is shifting toward targeted attacks.
- Payloads and increasing modularity are advancing stealthiness.
- Propagation speed is converging towards slower attacks with a stronger focus on sophisticated stealthiness.
- Propagation vectors include promising candidates for networks with homogeneous characteristics.
- Modular malware with second-channel payload types are now the state of the art.

While malware propagation is positively influenced by increasing speed, scope, and availability of attack vectors, these properties have a negative impact on all metrics that consider the stealthiness of malware. Detectability on the part of defenders, therefore, increases with the availability and successful application of anomaly detection that can find patterns and protocols used for propagation and for the C&C architecture [48].

8.1.3 Communication Patterns

Table 27 provides a classification with respect to detectable communication patterns. The table first scrutinizes discovery methods (scanning) utilized for mapping. It furthermore illustrates that some malware types do not scan networks. The recent development is shifting toward less conspicuous methods including highly distributed scanning, advanced topological preference scanning or alternative channels that do not require network scanning. This development may be the result of the fact that companies have started to deploy advanced network anomaly detection [48].

Malware uses several network and transport-layer protocols, which are discussed in Section 4.4.2. However, according to Table 27, today's malware implementations show a clear preference for the use of TCP as transport protocol, moving away from UDP. There has also been a notable shift toward encrypted communication. Since anomaly detection is expected to perform well in homogeneous infrastructures, several pieces of malware investigated here obfuscate C&C messages among native network traffic. See Section 4.4.5 for details on Regin's [94] obfuscation capabilities. Monitoring for the following malware activities can aid in network-based anomaly detection: [48]

- Forbidden communication attempts that highlight unsolicited connections which can be slowed or blocked.
- Scanning behavior, which indicates malware presence.

- Specific communication protocols that stand out in native traffic indicating the presence of malware.

Anti-malware tools can complement network-based malware detection with host-based methods when deployed on field devices.

We find it noteworthy that substantial differences in variants of C&C infrastructures are an indication that this may be a starting point for anomaly detection in predominantly homogeneous infrastructures. However, highly distributed structures as seen in [31, 175] may be difficult to detect. At this point, many malware instances still use standard protocols but encrypt all C&C traffic. Hence, anomaly detection may discover unsolicited traffic, which can be slowed or blocked. There are historic examples [29] in which covert channels were used for C&C, yet modern malware currently resorts to encryption instead of stealth tactics. The vast differences in the architectures and communication patterns of C&C structures seem to provide enough stealthiness to hide such traffic from typical heuristic comparison algorithms [48].

Based on Table 27, the following findings can be seen: [48]

- Network discovery is shifting toward complex low-visibility, high-distribution, modern topological or none-scanning methods, and away from aggressive blind-scan and simple topological-scan methods.
- Malware increasingly obfuscates C&C communication streams among native network traffic.
- Instead of using covert techniques, malware communication is usually encrypted as the vastness of communication patterns seems to provide enough obfuscation that such traffic is overlooked.
- There is a clear preference for TCP over UDP.
- Although standard protocols are used, C&C traffic is mostly encrypted and obfuscated. Hence, modern anomaly detection may discover unsolicited traffic.
- Anomaly detection is expected to perform well in homogeneous infrastructures to detect the selective use of protocols, forbidden communication, and scanning behavior.
- C&C is generally not highly distributed, thus, routed through many nodes for obfuscation reasons. However, there exists some examples for highly distributed C&C structures that are challenging to detect.

Table 27: Malware communication patterns, as presented in [48]

Name	Discovery	Protocols	C&C Method	Distributed C&C
CodeRed1	Blind scan	TCP	None	No
CodeRed2	Topological scan	TCP	None	No
Nimda	Blind scan	TCP, UDP	None	No
Slammer	Blind scan	UDP	None	No
Sality	Distributed Blind	TCP, UDP, HTTP	P2P	No
Conficker	Topological scan	TCP, UDP, SMB, HTTP	DNS flux, HTTP, P2P	Yes
Regin	Topological scan	TCP, UDP, HTTP, HTTPS, SMB	HTTP, HTTPS, SMB	Yes
Aurora	None	TCP, HTTPS, TLS, Email	WinAPI	Yes
Stuxnet	Topological scan	TCP, HTTP	P2P, WinAPI	No
Duqu	Topological scan	TCP, HTTP, Email	P2P, Obfuscation [1]	No
Flame	Topological scan	TCP, HTTP	HTTP, USB	No
Gauss	None	None	USB, HTTP	No
Duqu2	Topological scan	TCP, HTTP, HTTPS, Network pipes, Email	Network pipes, HTTP, Mailslots, Obfuscation [1]	No No
Equation	None	None	USB	Yes
BlackEnergy3	Topological scan	TCP, HTTPS, Email	HTTPS	No
Cozy Duke	Topological scan	TCP, HTTP, HTTPS, Email	HTTP, HTTPS	No
AdWind	None	HTTP, Email	TLS	No
Locky	None	HTTP, Email	IP flux, HTTP	Yes
PLC-Blaster	Topological scan	TCP, S7CommPlus	TCO	No

Notation: (1) Obfuscating C&C among Standard traffic by choosing native protocols.

8.1.4 Persistence and Defensive Mechanisms

Table 28 discusses the defense mechanisms of malware that increase its persistence against removal efforts. Morphism, as discussed in Section 4.4.4, denotes the ability of malware to autonomously change and adapt its appearance, including bit patterns in memory and for communication. A malware's morphism type can be used as a metric that characterizes its stealthiness and self-defense mechanisms. The payload structure has moved away from monomorphic toward polymorphic types, that represent by far the largest group in our sample. There is only one metamorphic example, i.e., AdWind [72, 89], which shows promising obfuscation features, but only in homogeneous environments. AdWind runs in JAVA and is capable of installing its native environment on multiple platforms. Future developments may, however, shift toward metamorphic types as AdWind is well known for its difficulty to detect and prolonged persistence although it uses no defense mechanisms other than recompilation and encryption on a local host [48].

Evasion of anti-malware tools was not a common feature in older malware. However, as malware's complexity increases, the huge effort spent in development is a high incentive to hide such evolved malware. Modern malware often evades and more recently even dismantles anti-malware tools by injecting malicious code into running system processes, effectively rendering them useless against a particular threat. Should a version of anti-malware tools be installed that cannot be counteracted, retreat and cleanup tactics are often used in order to protect the intellectual property such advanced malware represents. [48]

Several malware types are also memory residents that are altogether unreachable for common anti-malware tools. There are a number of ways of residing in memory or even partially on the hard drive in encrypted form, decrypting only the parts that are required into memory [48].

Aside from encryption, most modern malware types also obfuscate their code, making them even less detectable by heuristic-based anti-malware tools. This behavior is characteristic for polymorphism, cf. Section 4.4.4, and goes as far as to use heavy multi-layer obfuscation with encryption and periodic reiteration, e.g., metamorphism [72, 89]. This is an indication that malware authors do not yet perceive the need for such heavy defenses which indicates an advantage for attackers, as the technology for obfuscation exists already but is not yet widely used [48].

Service injection enables malware to survive restarting the host, which is an indication that increasing malware complexity supports greater persistence. Most sophisticated types are capable of hijacking running drivers, maintaining their functionality while obtaining autostart capabilities. Duqu2 [96], as

Table 28: Malware persistence and defense-mechanisms, as presented in [48]

Name	Morphism	Evasion	Memory residency	Code obfuscation	Service injection	Uninstall mechanism	Rapid reinfection	Cleanup [4]
CodeRed1	Monomorph	–	✓	None	–	–	✓	–
CodeRed2	Monomorph	–	–	None	–	–	–	–
Nimda	Monomorph	–	–	None	–	–	–	–
Slammer	Monomorph	–	–	None	–	–	✓	–
Sality	Polymorph	✓	✓	Simple	✓	–	–	–
Conficker	Polymorph	✓	✓	Complex [3]	–	✓	✓	–
Regin	Polymorph	–	–	Simple	–	–	–	–
Aurora	Polymorph	✓	–	Simple	✓	–	–	–
Stuxnet	Polymorph	✓	✓	Simple	✓	✓	–	✓
Duqu	Polymorph	✓	–	Simple	✓	–	–	✓
Flame	Polymorph	✓	–	Simple	✓	–	–	–
Gauss	Polymorph	✓	–	Simple	✓	✓	–	✓
Duqu2	Polymorph	✓	✓	Simple	–	–	✓	–
Equation	Polymorph	✓	✓[2]	Simple	✓	–	–	–
BlackEnergy3	Polymorph	✓	–	Simple	✓	–	–	–
Cozy Duke	Polymorph	✓	–	Simple	✓	–	–	–
AdWind	Metamorph [1]	✓	✓[2]	Complex [3]	–	✓	–	✓
Locky	Polymorph	–	–	Complex [3]	✓	✓	–	✓
PLC-Blaster	Monomorph	–	–	None	–	–	–	–

Notation: (✓) yes, (–) no

(1) Utilize recompilation, padding, obfuscation, multi-layer encryption.

(2) Decrypt modules into memory or stores them in encrypted form.

(3) Use complex multi-layer code obfuscation and encryption.

(4) Impede malware detection and analysis.

a notable example, shows no client-side persistence. Using an alternative approach, it resides in local servers and rapidly reinfects hosts after the boot sequence [48].

Finally, clean-up mechanisms, e.g. self-uninstall, are rarely found. As discussed in Section 4.4.8 there are variants capable of disinfection local drives upon triggers such as time triggers, or successful target detection, or on specific C&C commands [48].

Based on Table 28, the following findings can be seen: [48]

- Development has moved away from monomorphism toward polymorphism. Some metamorphic types exist that are acknowledged for their prolonged persistence and may represent an indication of future developments [89].
- The increasing complexity in advanced malware provides an incentive to protect its intellectual property, thus, malware developers may start implementing better encryption and obfuscation algorithms.
- The evasion of anti-malware tools is extensively implemented and should be considered the state of the art.
- Several malware types are capable of memory residency, increasing their persistence. However, there are other effective methods to achieve persistence using local storage.
- Traditional methods such as heuristics are becoming less efficient at detecting malware yielding their dominance to anomaly detection.
- Persistence through service injection must be considered to be the state of the art.
- Neither rapid reinfection nor cleanup mechanisms are yet widely deployed.

8.2 Future Evolution of Malware for Smart Grids

This section considers the evolution of Internet malware targeting smart grid environments. We expand on the perspective we opened in Section 2.2.7 and provide specific examples as to what future smart grid malware may look like, considering differences between Internet and smart grid communications. We first extract the specific novelties and strengths that existing types provide. This way, we gain insight into functionalities that might be used in construction kits for future malware. Then, we examine trends and summarize features that we expect to evolve. We elaborate on attack vectors specific to future smart grid environments in Section 8.3 and discuss defensive strategies in Section 10. Table 30 maps the attack scenarios corresponding to the effective defensive measures. Based on this, we define three novel hypothetical malware types suited for smart grid environments. The first step in designing advanced (hypothetical) novel malware for future smart grids is to determine the most highly evolved features that can be found in existing malware. These outstanding capabilities are summarized in Table 29. However, the investigated examples are not limited to these features and we expand on them for future attacks in the following sections [48].

We list future *trends we expect to see* concerning both the Internet and smart grids, however, we focus on the latter. These include: [48]

- Decreasing importance of propagation speed to the benefit of stealthiness and prolonged persistence in networks.
- Evolution of malware to evade signature-based detection methods, by:
  - Metamorphism replacing modern polymorphism.
  - Sophisticated multi-layer encryption.
  - Complex code obfuscation.
  - Unobtrusive network scanning methods.
  - Obfuscation of propagation in native network-traffic and increased passive propagation.
  - Covert and/or distributed C&C communication.
- Increasing versatility and complexity of anti-detection mechanisms beyond signature-based detection to the benefit of persistence and stealthiness in order to defeat anti-malware tools through the use of:
  - Rapid local recompilation.
  - Sophisticated multi-layer encryption.
  - Complex code obfuscation.

Table 29: Distinct strengths of malware, as presented in [48]

Name	Distinct strengths
CodeRed1	Small size (4 kB) and speed of spreading
CodeRed2	Designed to spread much faster than CodeRed1 using a localized scanning strategy
Nimda	Small size (60 kB) and speed of spreading
Slammer	Extremely small file size (404 byte) and even higher speed of spreading than CodeRed2
Sality	Highly distributed scanning by $>$ 3 million hosts
Conficker	Patches known vulnerabilities only to use them as backdoor
Regin	Only utilizes VPN between modules, even locally
Aurora	Watering hole attack on secondary targets, in order to gain access to primary target
Stuxnet	Physical destruction, stealthiness & prolonged persistence
Duqu	Infects internal processes to dismantle anti-malware tools
Flame	Hosts a fake Windows update-service in order to spread
Gauss	Silent on networks, highly covert, slow spreading
Duqu2	Rapid reinfection by unusual S2C-vector without any persistence on the hosts
Equation	Silent on networks, infects hard drive firmware
BlackEnergy3	Uses legal signatures, physical destruction
Cozy Duke	Watering hole attack to steal credentials
AdWind	Highly obfuscated, uses multi-layer encryption and utilizes local recompilation - metamorph
Locky	Obfuscation, encryption and disinfection features
PLC-Blaster	Selectively targets Siemens industry PC's

  - Increased use of memory residency or other effective means to avoid anti-malware tools.

- Increasing modularization & customization adding flexibility to all areas of the generic model introduced in Section 4.4.
- Increasing targeted attacks, targeting cyber-physical systems due to their excellent potential as leverage against companies or communities.
- Increasing profitability of zero-day vulnerability research and increasing number and complexity of access and propagation vectors.

8.3 Attack Types and Mitigation Specific to Smart Grids

This section elaborates on potential attack methods, we consider feasible in the near future, specific to smart grid environments and discusses possible mitigation techniques, numbered in accordance with Table 30 which consolidates them. As mentioned in Section 4.4.6, we distinguish between five general attack goals.

8.3.1 Disruption Attacks

Disruption attacks aim to suppress a service for a period, leaving it unavailable for regular operation.

(a) *DDoS-attacks from outside, targeting inside assets (Inbound attacks)*: Such attacks can result in delays, outages, and monetary loss. Examples have been discussed in [23, 114, 131, 147, 166]. Targets inside smart grids could be, e.g., meter-data-management-servers, key-management-servers, email-servers, network storage, and critical firewalls.

These attacks should be filtered at the perimeter firewalls, thus, be covered with baseline security measures (D1). The most extreme cases could force utilities to rent a traffic-scrubbing service for a time, alleviating traffic off their local firewall.

(b) *DDoS-attacks from inside targeting inside assets (Internal attacks)*: Such attacks can originate from compromised assets inside the utility network and can target central or high level components e.g., control centers. The attacks may result in outages, depending on the resilience of the system and fallback strategies in place.

Inside attacks should be filtered by internal firewalls between subnetworks, thus are covered in the baseline security measures (D1) within the segmentation measures. Additionally, service reduction may help identifying the infected hosts.

(c) *DDoS-attacks from inside attacking targets outside (Outbound attacks)*: Compromised assets, e.g., smart meter networks, represent a substantial number of devices that can be used to run outbound DDoS-attacks.

Outbound DDoS should be mitigated by perimeter firewalls (D1, D7), which identifies the infected hosts. For this, the services should be reduced to the necessary amount, thus, no, e.g., email or print services should be active in critical networks.

(d) *DDoS-attacks on certain user groups (Selective harassment)*: Rather than targeting central assets to disrupt all service the selective harassment of user groups through abuse of remote-disconnect features on compromised smart meters can damage the reputation of utility companies.

Selective DDoS attacks can be mitigated by baseline security measures (D1) such as internal firewalls, while identifying infected hosts, aided by anomaly detection (D3), a reduced services environment (D6), and content filtering (D7). Initial infection of smart meters can be prevented by physical security measures (D4), or prevent the use of removable media (D5).

8.3.2 Destruction Attacks

There are several precedents from the recent past, cf. Section 4.4.6, in which industrial equipment was targeted and destroyed, delaying production considerably. The affected process has to be rebuilt, which takes a long time. The difference between a disruption and a destruction attack is that once the disruption attack ends, e.g., DDoS, the equipment recovers and becomes reusable again, whereas after a destruction attack the affected precesses need to be restarted or even rebuilt to recommence operation. Potential use cases include, but are not limited to:

(a) *Disconnect households:* Compromised smart meters with remote power-disconnect features may drop many households at once, aggravating cascading effects or blackouts in the local power grid. Furthermore, power reconnect features can further destabilize the power grid by oscillate-switching households, negatively impacting grid stability.

    Disconnection attacks can be prevented by physical security measures (D4) and a reduced services environment (D2, D6), by well configured firewalls (D1, D7), network segmentation (D1), and anomaly detection measures (D3) to identify infected hosts.

(b) *Destroy power management:* Central power management and switching equipment can be attacked and destroyed, as demonstrated in the Ukraine incident [113]. Details about access methods are available in Section 4.4.1 and on the attack vectors in Section 4.4.6.

    This attack may be mitigated by baseline security measures (D1) in combination with a trusted and controlled environment (D6) with a reduced number of services (D2) and network segmentation (D1). Content filtering (D7) and anomaly detection (D3) measures may also be effective against an attack similar than in the Ukraine. Physical security (D4) and the prevention of using removable media (D5) can stop an attack in its initial phase, while long term backups (D9) can help in the rebuilding process.

(c) *Influence critical electrical nodes in the field:* The targeting of power-switching equipment in the field can be optimized as demonstrated in the examples devised by Wang et al. [183] and Rosas-casals et al. [150]. They introduce metrics for calculating critical electrical nodes, relieving effort on the attacker's side. In combination with DDoS or targeted cyber-attacks potent attack vectors are opened.

    All security measures mentioned in the security baseline (D1), a reduced service environment (D2), and anomaly detection (D3) can mitigate these attacks, while physical security (D4) and a secure environment (D6), e.g., segregated networks can prevent intrusions.

(d) *Modify sensor data:* Compromised sensors or man-in-the-middle attacks on sensor data may generate false decisions in the control center, leading to blackouts.

    Modified sensor data can be prevented by physical security measures (D4) including event logging in a trusted environment (D6). Additionally, baseline security measures (D1) and leakage protection (D8) can mitigate ongoing attacks.

(e) *Tamper with clock synchronization:* The time synchronization and power measurement data on sensors, e.g., PMUs, can be manipulated to drift apart, triggering emergency switching [86].

    Manipulated PMU data can be prevented by physical security (D4) and logging in a trusted environment (D6). Baseline security measures (D1), anomaly detection (D3), and leakage protection (D8) can mitigate ongoing attacks.

8.3.3 Data theft

Stealing a commodity, e.g., information, affects the target indirectly by its reputation. Information revealed to competitors or leaking critical information about the network topology or credentials can improve attack efficiency and future attack preparation. Data theft (espionage) are among the most common attack types according to [95, 96, 98, 123, 158, 169, 175].

Generally, our investigations revealed that anomaly detection (D3) in a trusted network environment (D6) and proper leakage protection (D8) through firewalls with content filtering (D7) could mitigate typical data theft attacks. Physical security (D4) and forbidden removable media usage (D5) can prevent initial entry vectors. Baseline security measures (D1) should be implemented to create a minimum level of security that can partially prevent these data theft attacks.

(a) *Espionage:* Eavesdropping on user data, e.g., power consumption or billing data, collect information on persons or groups, especially when complemented with Internet data.

(b) *Ruin credibility of users:* Legal actions may be forced upon victims, effectively occupying their time, e.g., if they become suspect after an attacker modifies meter readings or profiles.

(c) *Ruin reputation of providers:* Satisfaction ratings of utility companies can be negatively impacted by stealing information and manipulating billing.

(d) *Ruin reputation of manufacturers:* Satisfaction ratings of manufacturers can be affected via data manipulation and information theft, leading utility companies to reconsider future investments.

(e) *Sell long term data:* Historical consumption data can be sold for statistics, e.g., to competitors.

(f) *Sell live data:* Live energy consumption data can be sold for home intrusion optimization.

(g) *Market manipulation:* Energy markets can be manipulated by compromising smart meter networks with switchable loads through exfiltrated information. Additionally to the mitigation measures above, we add local service reduction (D2) which can further increase the detectability of attacks. Long term backups (9) can aid in recovery processes.

(h) *Bill manipulation (Attack as a service):* Attackers can offer reduced energy bills as a service, effectively manipulating billing with false energy data, which represents stealing from a utility. Additionally to the mitigation measures above, we argue that local service reduction (D2) can increase the detectability of such attacks. Long term backups (9) can aid in recovery processes.

8.3.4 Extortion Schemes

Extortion schemes attempt to demand ransom for releasing a captured commodity or service [9, 162].

(a) *Threat of destruction:* Compromised smart meters harbor the potential to affect power grids. Extortion of utility companies may be used as a prerequisite for destruction attacks, yet may also reveal the attackers' presence. We assume that extortion schemes are conducted alongside a real threat instead of a bluff.

Baseline security measures (D1) and content filtering (D7) in a trusted environment (D6) may increase the chance that anomaly detection (D3) can discover the source of such an attack. Furthermore, physical security (D4), leakage protection (D8), and limited interfaces (D5), e.g., removable media, can prevent it in the initial stages. Long term backups (D9) can be used to recover the infected hosts.

(b) *Threat of DDoS:* DDoS attacks can nowadays be accompanied by ransom notes. In the case of critical infrastructures cascading effects may serve as excellent leverage.

Baseline security measures (D1) and content filtering (D7) should suffice to mitigate DDoS attacks, thus, the threat of such may be acceptable for risk management considerations.

(c) *Crypto-locker:* The infiltration of crypto-locker malware into utility networks may lead to devastating data loss through the encryption of important files.

Cryptolocker infections can be prevented by limiting the input interfaces (D5), and leakage protection (D8). However, baseline measures (D1) alongside a trusted environment (D6) with controlled and reduced services (D2) can level the ground for effective anomaly detection (D3). Additionally, proper recovery procedures (D9) will be required from the data- and image-backup concept.

8.3.5 Repurpose Attacks

Repurposing includes all methods that change the behavior of a host from its intended function.

(a) *Fake servers:* Infected hosts can act as fake update servers as demonstrated by Flame [97] in order to increase the spreading of the infection. This method is also conceivable in smart grid networks.

The prevention of the initial intrusion should be sufficiently covered by baseline security measures (D1) in combination with limited interfaces, thus, physical security measures (D5) on the hosts. Furthermore, a trusted environment (D6) with reduced services can aid anomaly detection (D3) measures to discover the attackers and the infected hosts. Backups (D9) should be kept for recovery purposes.

(b) *Proxies:* Infected hosts can be used as proxies to obfuscate C&C.

The initial attack vector can be prevented with baseline security measures (D1), a reduces services environment (D2), and anomaly detection (D3). A trusted environment (D6) behind a content filter and perimeter firewall (D7) can prevent further propagation. regular backups (D9) will be required for recovery purposes.

(c) *Distributed computing:* Compromised smart meter networks host numerous devices that can be re-purposed for distributed computing, e.g., Bitcoin mining, or brute forcing hashes.

    Initial attack vectors can be prevented by baseline security measures (D1) and perimeter firewalls with content filtering (D7). Anomaly detection (D3) may be able to identify the infected hosts in a trusted environment (D6) with only the necessary services active (D2). Backups (D9) can be used to restore infected hosts.

Most attacks mentioned above, are hypothetical yet are based on the increasing complexity of malware technology, cf. Tables 25 through 28. Sophisticated stealthiness and persistence can be highly effective in mapping targets. Extensive data theft campaigns, therefore, usually precede cyperphysical attacks [56, 58, 123, 150].

Table 30: Mapping applicable security measures to attack scenarios, as presented in [48]

		Attack types (Section 8.3)																							
		Disruption				Destruction					Data theft								Extortion			Repurpose		
		a	b	c	d	a	b	c	d	e	a	b	c	d	e	f	g	h	a	b	c	a	b	c
Defense, cf. Sections 4.3 & 10	(D1) Baseline security measures	✓	~	✓	✓	✓	~	~	~	~	~	~	~	~	~	~	✓	✓	~	✓	~	~	~	~
	(D2) Reduction of service	–	~	~	–	–	~	~	–	–	–	–	–	–	–	–	~	~	–	–	~	~	~	~
	(D3) Anomaly detection [1]	–	–	–	~	~	✓	~	~	~	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	~	✓	✓	✓
	(D4) Physical security	–	–	–	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	–	–	~	–	–
	(D5) Removable media	–	–	–	~	~	~	~	–	–	~	~	~	~	~	~	~	~	~	–	~	~	–	–
	(D6) Trusted environment	–	–	–	~	~	✓	~	~	~	✓	✓	✓	✓	✓	✓	✓	✓	~	–	~	✓	~	~
	(D7) Content filter and firewall	✓	~	✓	~	~	~	~	–	–	~	~	~	~	~	~	~	~	~	✓	–	~	✓	~
	(D8) Leakage protection	–	–	–	–	–	✓	~	✓	~	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	~	–	–	–
	(D9) Backup / recovery	–	–	–	–	–	✓	✓	–	–	–	–	–	–	–	–	✓	✓	✓	–	✓	~	~	~

Notation: (✓) Effective, (~) Partially effective, (–) Ineffective, (1) Effective in combination with measures invoked upon detection.

9 Malware-Attack Simulation: Results and Discussion

> ***Notice of adoption from previous publications:***
> *Parts of this chapter have been previously published in [46]. We introduce simulation results on malware propagation and attack duration for mesh and cell topologies. The authors main contribution was the development and participation of the topological model and the malware models. The co-authors contributed benefit on plausibility, correctness, and structure.*
>
> *[47] discusses the implications monocultures can present in critical infrastructure networks.*

We discuss the simulation results of simulation performance, network segmentation, and the attack defendability of the four developed network topologies against our three proposed different malware types.

9.1 Capabilities of the Simulation Environment

In this section we test the capabilities of the simulation environment and elaborate on results that we use to show how the model performs under different circumstances.

9.1.1 Dependency of Infection Time over Increasing Network Size and Increasing Node Distance in Single-Mesh Networks

As discussed in Section 5, we utilize the ns3 simulation environment and the OLSR mesh network protocol. All parameters used in these simulations can be found in Section 5.6, cf. Table 17.

First, we discuss the effect of increasing wireless distance with simulations of different-sized mesh networks. Figure 26 shows the infection durations required to infect all nodes in each mesh network. Furthermore, it shows the increasing network size, while the colors illustrate wireless node distances between each node. All simulations are conducted within one mesh network with the goal to test the limits of our simulation environment. The first observation is that wireless distance has no effect on the infection time in those single-mesh networks smaller than 80 nodes. This can be explained due to increasing control traffic that produces delays in the network, consuming the bandwidth. Therefore, we set our node limit of 80 nodes per mesh network. Furthermore, a maximum size to 80 nodes is also consistent with the discoveries of Palma et al. [143] and Audeh [7]. They argue that large mesh networks degrade in performance due to increasing control traffic

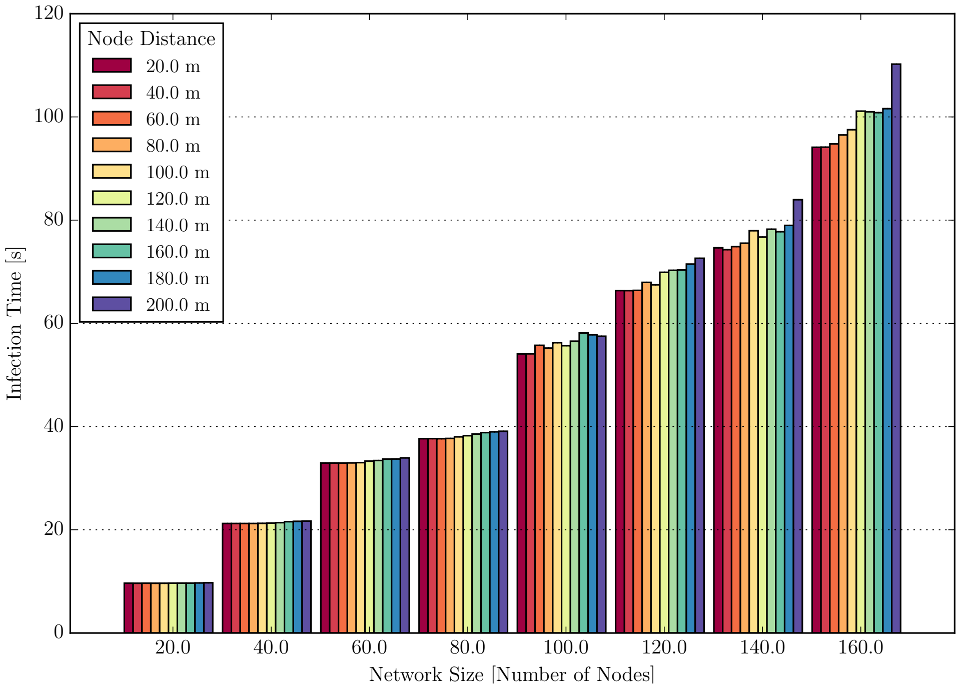


Figure 26: Dependency of infection time over increasing network size at increasing node distance

that ultimately consumes the networks bandwidth. The results in Figure 26 support this hypothesis. Furthermore, we define a wireless node distance of 108 meters, as per the mean value of our parent dataset in Section 3.6.2, as our standard distance between nodes.

We include the following simulation settings:

- Native legitimate active traffic of the network nodes is activated.
- Native traffic intervals increase dynamically according to network size.
- Malicious payload size is fixed to 500 bytes.
- Exploit time for infection simulation is set to 10 ms.
- All random factors are deactivated, cf. Section 5.3.
- OLSR converge time is set to 3 cycles for settling initial control traffic.
- Malware type is set to endemic malware, cf. Section 5.5.

Table 31: Simulation performance settings

Scenario name	Number of mesh networks	Number of nodes	Total number of nodes
f1m	1	3000	3000
f13	30	100	3000
f65	50	60	3000
s12	20	100	2000
s62	25	64	1600
s43	36	49	1764
M1	1	10-80	10-80
M10	10	10-80	100-800
M20	20	10-80	200-1600
M30	30	10-80	300-2400
M40	40	10-80	400-2000
M50	50	10-80	500-2500
N10	1-60	10	10-600
N50	1-40	50	50-2000

9.1.2 Simulation Performance of Large Multi-Mesh Networks

We discuss the results of testing the limits of our simulation model in this section. Figures 27 and 28 illustrate an overview of several simulation-sets that are discussed in the following paragraphs in more detail. The aim of these heat-maps is to outline the maximum performance of the simulation environment and what size of networks can be accommodated. We include simulations that represent our maximum tests with 3000 nodes, simulation sets that represent our mesh network topologies in a symmetric setup, e.g., 5*5 meshes per network, simulation sets in which we increase the number of mesh networks (M), and simulation sets in which we increase the number of nodes (N).

Based on our results in Section 9.1.1, our initial tests in this section, and the results of Audeh [7] and Palma et al. [143] we set the maximum number of nodes per mesh network to 80 nodes. Furthermore, we set the maximum number of mesh networks that can be simulated to 50 due to the results in this section.

Additionally, we include several single simulation scenarios that have two fixed values, i.e., mesh size and number of nodes, namely f1m, f13, f65, s12, s62, and s43. The following paragraphs and Table 9.1 detail on each simulation-set, why they have been chosen in their respective number of nodes and mesh networks, their names, and settings. Several more scenarios, namely, M1, M10, M20, M30, M40, and M50 illustrate the models performance with a fixed number of mesh networks and an increasing number of nodes. All simulations are illustrated in the Figures 29 through 34 and in the heat-maps, cf. Figures 27 and 28. The heat-maps illustrate, in vertical orientation the node numbers per mesh network, and in horizontal orientation the number of mesh networks. Finally, two scenarios that show the models performance in vertical orientation in the heat-maps are tested, namely N10 and N50. See Figures 35 and 36 for details.

System Boundaries

We define the term "High Resource Consumption" as the system border of our simulation environment. At this point, either the elapsed simulation time exceeds 55 hours per successful simulation or the network-model seizes operation when control traffic exceeds the network bandwidth.

For our simulations we utilize a dedicated simulation PC with Ubuntu Linux 16.04 LTS, the ns3 simulation environment (ns-3.26), an Intel Core i7 processor, 16 GB Ram, and 7200 rpm HDD drive. We decided to use a dedicated PC instead of a simulation cluster because the ns3 network environment can only be used in multi-core mode with P2P networks. Mesh networks are excluded from multi-core processing.

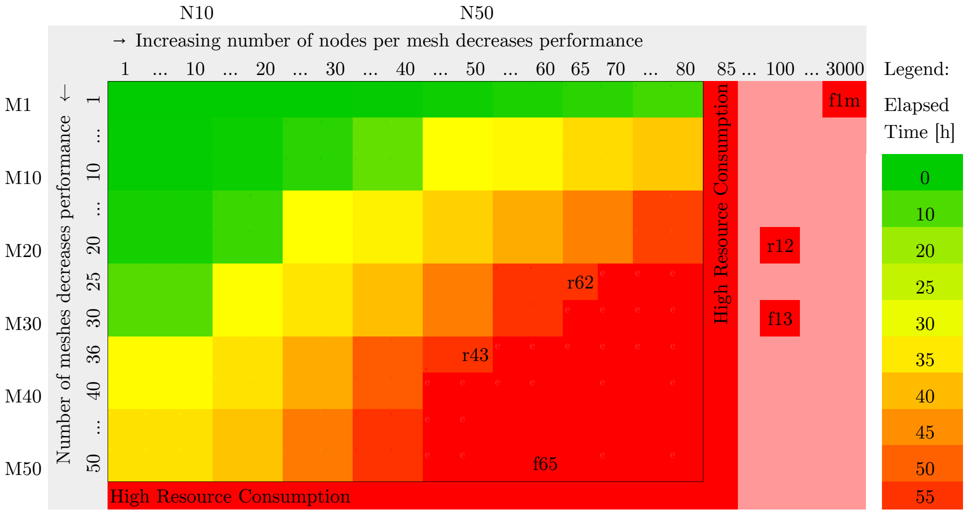


Figure 27: Simulation performance, heatmap elapsed simulation time

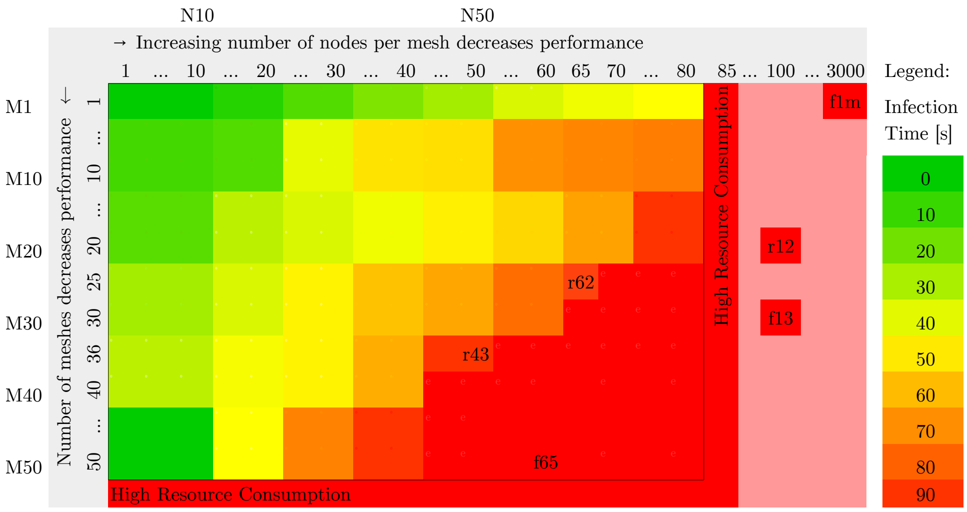


Figure 28: Simulation performance, heatmap simulated infection time

Scenario f1m

This scenario simulates a single mesh network that includes 3000 nodes. However, the simulation was canceled after 400 hours upon no visible progress. We name it "full, 1 mesh" (f1m).

Scenario f13

This scenario includes 100 nodes per network over 30 mesh networks, thus, 3000 nodes. We canceled the simulation after 338 hours upon no visible progress. We name this scenario "full, 100 nodes, 30 mesh networks" (f13).

Scenario f65

This scenario includes 60 nodes per network over 50 mesh networks, thus, 3000 nodes. We canceled the simulation after 212 hours upon no visible progress. We name the scenario "full, 60 nodes, 50 mesh networks" (f65).

Scenario s12

This scenario considers a smaller node-set, i.e., 100 nodes over 20 mesh networks, thus, 2000 nodes. The simulation was successful after more than 60 hours elapsed simulation time. Full infection was achieved after 1 minute 53 seconds. We name this scenario "small, 100 nodes, 20 mesh networks" (s12)

Scenario s62

This scenario considers a smaller node-set, i.e., 64 nodes over 25 mesh networks, thus, 1600 nodes. The simulation was successful after 24 hours and 42 minutes elapsed simulation time. Full infection was achieved after 1 minute 3 seconds. We name this scenario "small, 64 nodes, 25 mesh networks" (s62)

Scenario s43

This scenario considers a small node-set of 49 nodes over 36 mesh networks, thus, 1764 nodes in total. The simulation was successful after 31 hours and 30 minutes elapsed simulation time. Full infection was achieved after 1 minute 9 seconds. We name this scenario "small, 49 nodes, 36 mesh networks" (s43)

Scenario M1

After testing fixed scenarios, we tested the model against a range of simulations. This simulation fixes the number of mesh networks but increases the number of nodes from 10 to 80 in steps of 10. *We name all following scenarios using a fixed number of mesh networks with "M"*. Increasing the node numbers, is repeated equally for all following scenarios. Figure 29 shows that the simulated infection time increases linear, whereas the elapsed simulation time exponentially, as expected in large simulation-sets. The simulated infection time is the time until all nodes are infected. The elapsed simulation time is the time the simulation takes in the real world. Some simulations take a very long time due to their network size compared to smaller scenarios.

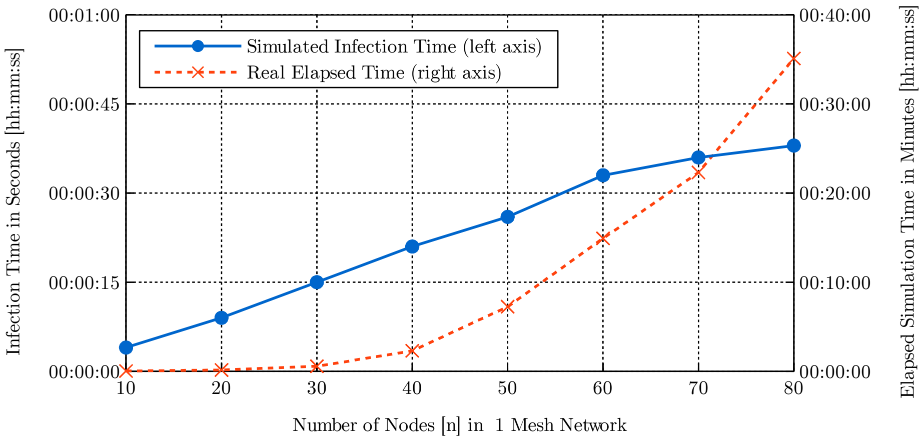


Figure 29: Simulation performance, scenario M1

Scenario M10

We increase the node number from 10 to 80. The total number of meshes is 10. The nodes multiply by the number of meshes, starting at 100 in the first and 800 in the last simulation. Results are illustrated in Figure 30.

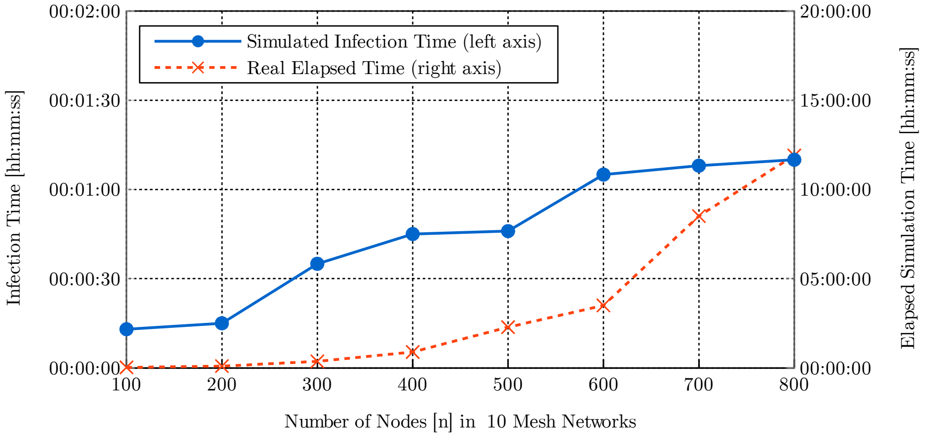


Figure 30: Simulation performance, scenario M10

Scenario M20

Appending on the previous two scenarios, we increase the number of mesh networks to 20. The minimum network size is 10*20 nodes, the maximum 80*20 nodes. As expected, infection time is nearly linear and the elapsed simulation time increases exponentially. See Figure 31 for details.

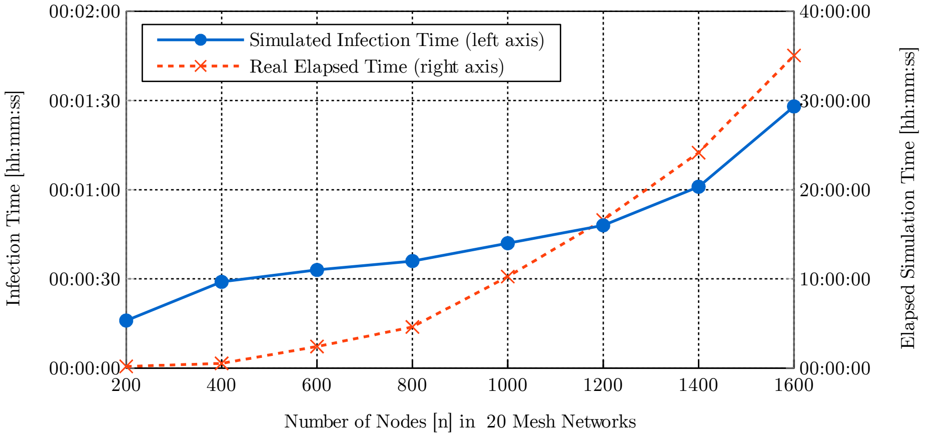


Figure 31: Simulation performance, scenario M20

Scenario M30

In scenario M30 we try to simulate network sizes greater than 1800 nodes at 30 mesh networks. However, the greatest possible simulation-set was 1800 nodes at 60 nodes per mesh and 30 meshes. The simulations with larger network sizes no longer converge, and the elapsed simulation time exceeds 37 hours. This is where the simulation model begins to break down due to increasing resource demand, and decreasing performance. See Figure 32 for details.

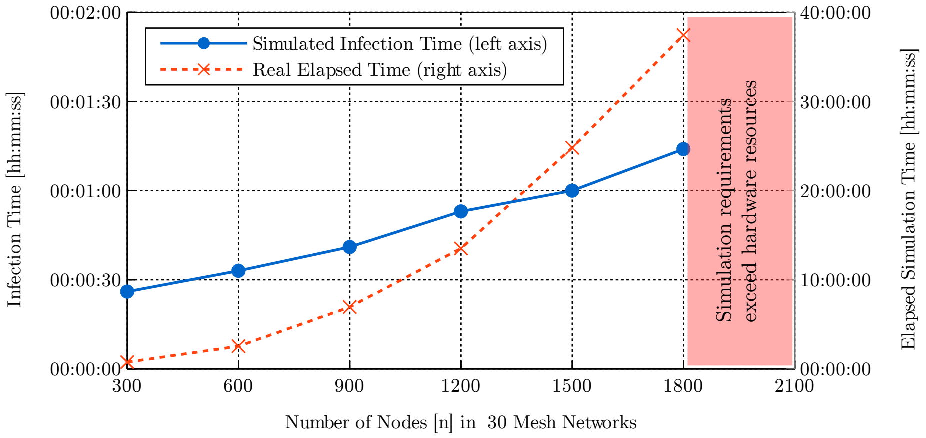


Figure 32: Simulation performance, scenario M30

Scenario M40

This set tests simulations on 40 mesh networks. We notice that above 1600 nodes (or 40 nodes in 40 meshes) simulations no longer converge. The elapsed simulation times exceed 30 hours. See Figure 33 for details.

Scenario M50

This set of simulations allows a maximum of 50 mesh networks, although most simulations no longer converge. See Figure 34 for details.

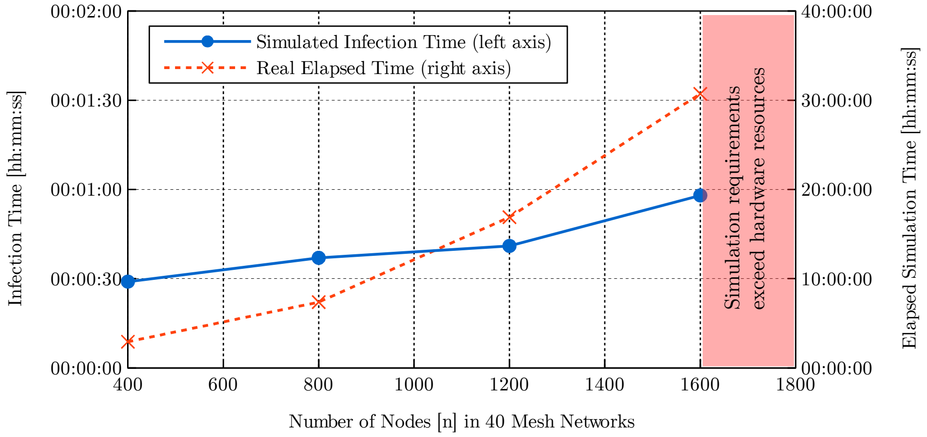


Figure 33: Simulation performance, scenario M40

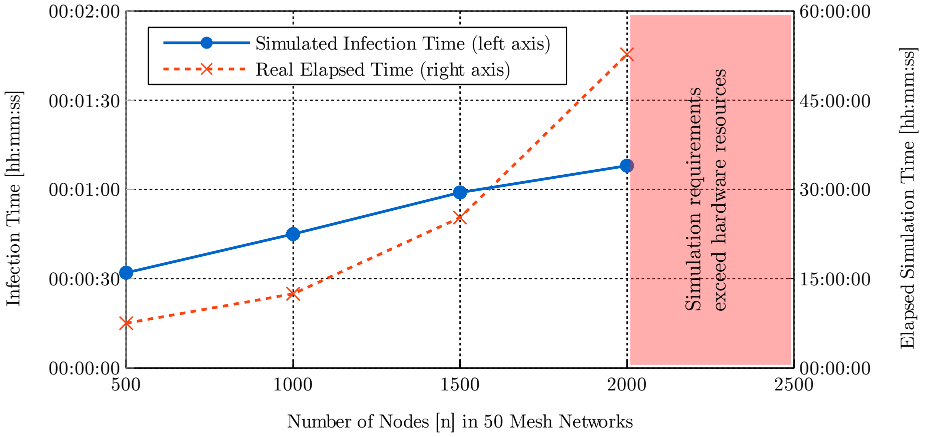


Figure 34: Simulation performance, scenario M50

Scenario N10

This simulation-set is illustrated in vertical direction in the Figures 27, and 28 and includes a fixed value of 10 nodes per mesh network at increasing number of meshes from 1 to 60 in steps of 10. Details are found in Figure 35. We test up to 70 mesh networks, however the maximum is met with 60 mesh networks. Since our limit is set at 50 mesh networks, we omit results with more mesh networks from the heat-maps.

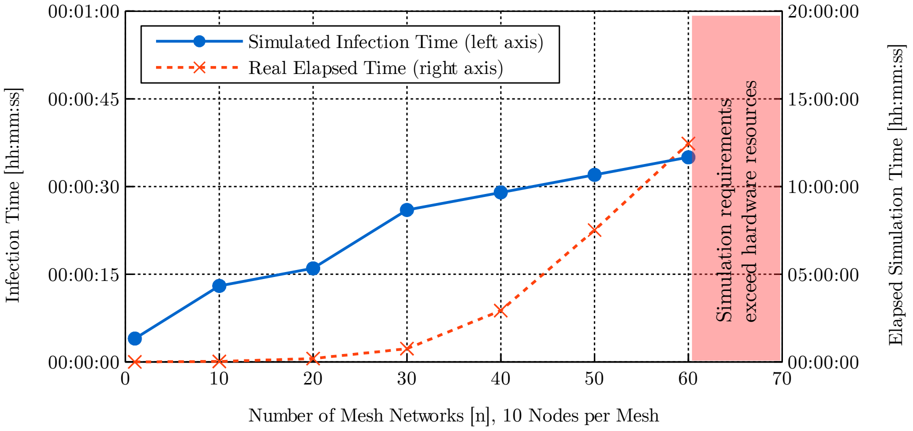


Figure 35: Simulation performance, scenario N10

Scenario N50

In the simulation-set N50 we set a fixed node number of 50 nodes per mesh and increase the number of mesh networks from 1 to 50 in steps of 10. See Figure 36 for details. The figure is limited to the maximum feasible network size. Therefore, we omit simulations larger than 30 mesh networks with 50 nodes per network.

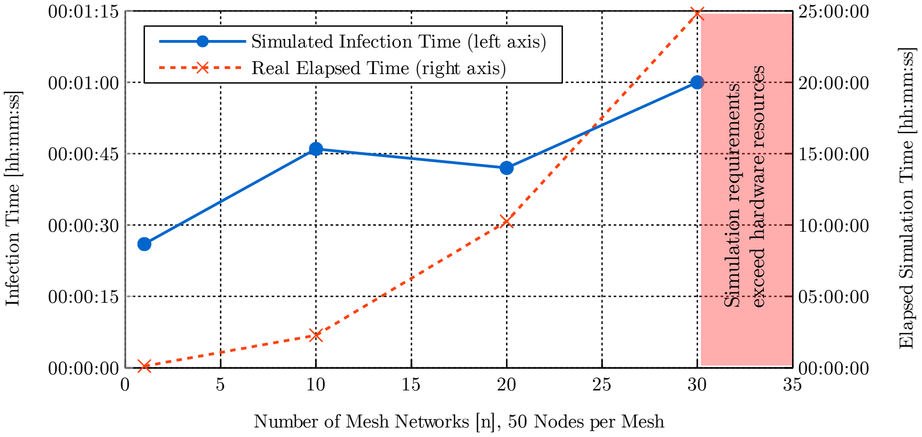


Figure 36: Simulation performance, scenario N50

Based on these initial results we find the statements by [7, 143] confirmed. Therefore, we limit our simulation model to a maximum of 70 nodes per mesh network and 10 mesh networks to retain good performance in further simulations.

9.1.3 Increased OLSR Control Message Cycle for Stationary Networks

As mentioned in Section 5.2.2 the OLSR control messages can be manipulated for longer update-cycles in stationary networks. We do not change our topology during simulation, thus, meet this requirement. Due to the limited scaling capabilities of OLSR we change the control message cycles from 5 seconds, as per [139], to 500 seconds for all simulations. Figure 37 illustrates a malware infection simulation with control messages set to the standard value. Figure 38 illustrates a malware infection simulation with manipulated control messages, showing decreased impact on the network performance. In the interest of decreased secondary effects from the OLSR protocol, because the nodes are stationary, we conduct all future simulations with the increased sending cycle. We assume future mesh-network protocols that support smart grid communications can solve the issue of overwhelming control traffic, as discussed in the outlook, cf. Section 11.7.

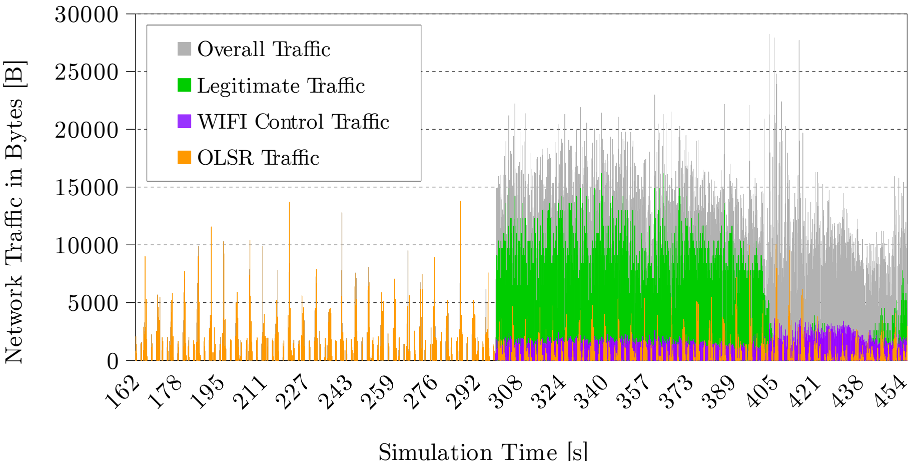


Figure 37: OLSR control messages with standard settings for small networks with scattered nodes

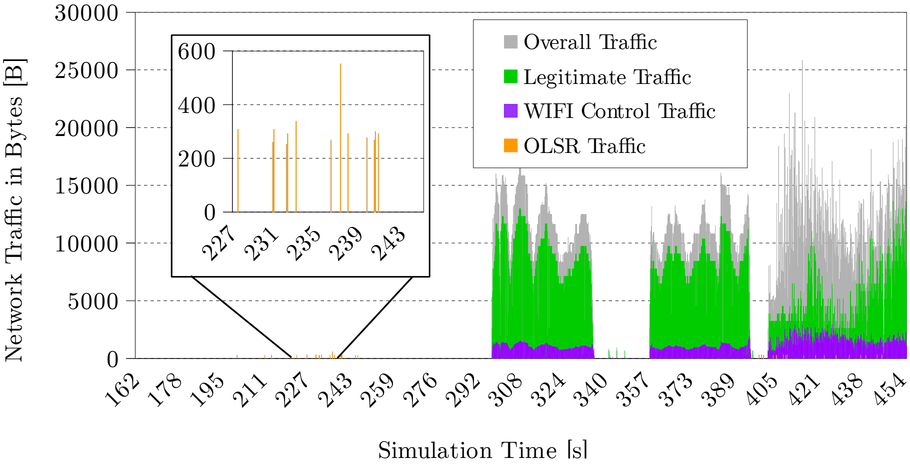


Figure 38: OLSR control messages with increased sending cycles for static and dense networks

9.2 Network Segmentation and Monocultures

In this section we discuss the failure of network segmentation measures and elaborate on the drawbacks of homogeneous infrastructures, i.e., monocultures, on an examplary simulation. The simulation is restricted to endemic malware, results and conclusions being applicable to other malware types, as detailed in [48]. First, network segmentation is introduced as an effective measure to contain malware propagation throughout this work. In case segmentation measures fail to protect the gateways, other parts of the network may be affected. Furthermore, monocultures can lead to identical vulnerabilities in a large set of devices even when residing in separate segments. Therefore, monocultures can provide attackers with an advantage to propagate. Figure 39 shows a simulation of two sub-networks of the same size, i.e., 49 nodes and a connecting gateway. We find that four key points in the time-line are of interest. These are:

Five key events in the simulation time-line in Figure 39 characterize any simulation run. Parameters that influence on the timing of these events include malware behavior, network size and network topology. When computing the timing of these events for distinct malware types and/or for distinct networks, the results can help to compare the performance of specific malware in specific network settings. In particular, the timings can help to rate and compare the robustness of specific network topologies when attacked by specific malware types.

These five key events are:

- *Patient zero:* Infection time of the first node, by definition at simulation start.

- *Infected gateway interface A:* Up to this point, infected nodes in network A are scanning for the gateway, infecting several local nodes in the process. Once the gateway is infected lateral propagation into network B is possible if the corresponding network interface of the gateway can be utilized. Already 20% of the nodes in network A are infected, i.e., 10% of all nodes in both networks.
- *Infected gateway inferface B:* The foothold in the target network B is established, i.e., the gateway is fully infected (interface A and B) and scanning in network B can commence.
- *Network A fully infected:* Full infection of network A is accomplished, i.e., 50% of all nodes in the whole setup. Assuming devices in network B to feature the same vulnerability, the malware already started infecting network B. In total (network A and B) 77% of all nodes are infected.
- *Network B fully infected:* Full infection of both networks (100% of all nodes) is achieved. All types of attacks (e.g., selective disruption or destruction, full disruption, etc.) can commence against both networks A and B.

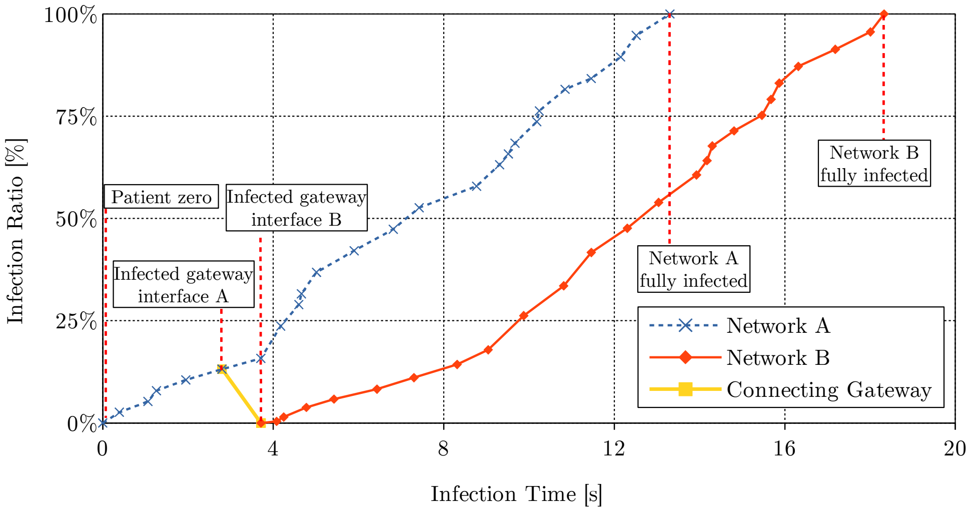


Figure 39: Network segmentation and monocultures, as presented in [47]

Monocultures can produce challenging vulnerabilities for a large number of devices, as seen in several cases [106,116]. Therefore, the protection of critical assets, such as a central gateway, is of utmost importance, serving as a last line of defense before malware can spread throughout the network or even to the control center.

9.3 Simulation Results and Attack-Defendability

In this section we discuss the simulation results for our fully equipped simulation environment based on the discoveries in Sections 9.1 and 9.2. In accordance with research questions 2 and 3, cf. Section 1.3, we elaborate on malware propagation inside smart grid networks. We illustrate the results by our three malware models, cf. Section 4.5, simulated over four network topologies, cf. Section 3.4. We use the vulnerabilities discussed in Section 4.6, our goal being to identify effective defensive measures against each malware type.

We focus our simulations on the most pressing attack scenario, namely the destruction attack, cf. Section 4.4.6. It requires the successful infection of either all ML-nodes, or the HL-node, which, in most cases is the fastest result. It can be accomplished quickly because only these neuralgic nodes must be infected. However, we also argue many other attack types within the simulation results because with progressing time, e.g., 75% of all nodes are infected, other attack types become possible. These other attack types do not represent such an immediate need for protection and could be considered as less challenging sub scenarios, cf. Section 11.7. Furthermore, we note that all nodes have a zero-day vulnerability, cf. Section 4.6, and reference again our network topologies, cf. Section 5.4. All parameters are available in Section 5.5.

We illustrate all simulations in an infection graph that was introduced in Section 6.3, and Figure 22. It shows the infection ratio over the infection time.

9.3.1 Centralized Topology - Simulation Results

The centralized topology is modeled as described in Section 5.4.1 with wired Point-to-Point (P2P) links between all hierarchy levels. Each LL-node is directly connected to the next overlaying ML-node in a star-configuration. ML-nodes have no local intelligence and only aggregate data for the HL-node in this configuration, as discussed in Section 3.4.1. LL-nodes cannot communicate laterally to other LL-nodes because this model uses a dedicated wired infrastructure with an FDMA separated PLC-uplink to the ML-nodes, and an WDMA separated optical fiber uplink to the HL-node. The expected results are fast infection of the ML-ICT and HL-ICT nodes as they represent the immediate communication destinations. After infecting the HL-node, which is the only decision maker, downward infection commences until all ML- and LL-nodes are infected. See Table 32 for a summary of the metrics.

Pandemic Malware

Figures 40 and 41 illustrate pandemic malware during the infection phase in a centralized topology. Infection commences very quickly and the first gateway is infected within 0.53 seconds, whereas the control center is infected after 0.62 seconds. The reason for fast infection in this P2P topology is that each node only has one communications partner, and the hierarchy levels are directly interconnected. This leaves little time for adequate reaction by human defenders, except for automated defenses. At this point the attacker has control over the HL-node, therefore, further propagation is not necessary for commencing destruction attacks by enabling remote power down sequences, which also affects all subjacent nodes. Should the attacker aim to commence another attack strategy besides destruction attacks, cf. Section 8.3, the following infection times apply. After 18.36 seconds the attacker has control over all ML-nodes, enabling spying on selected aggregate data, and after 18.82 seconds over 75% of all nodes, enabling the attacker to spy on a significantly large group which allows commencement of all attack types. After 49.44 seconds the attacker can selectively spy on all nodes, and commence all attack types.

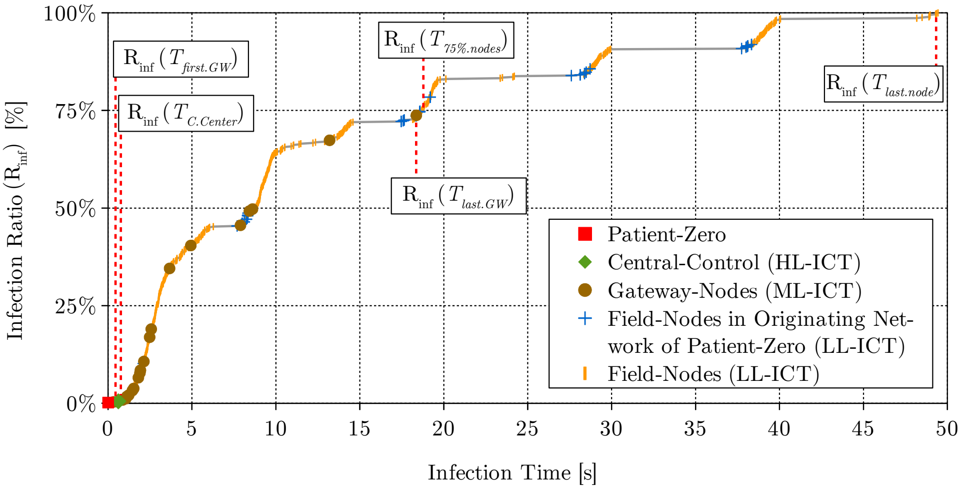


Figure 40: Pandemic malware in centralized topologies, infection graph

The pcap illustration in Figure 41 shows legitimate traffic, scanning traffic and malicious traffic inside one exemplary PLC link. We show this example to illustrate the extensive scanning taking place by pandemic malware. However, the malware has only one link available, yet it scans aggressively for other nodes, thus, transmits the payload once, trying to acquire other targets. This figure, representing one link, shows the infection of one node, thus, is exemplary for other PLC links in this topology. Aggressive scanning results in a large number of redundant scans exhausting the link bandwidth.

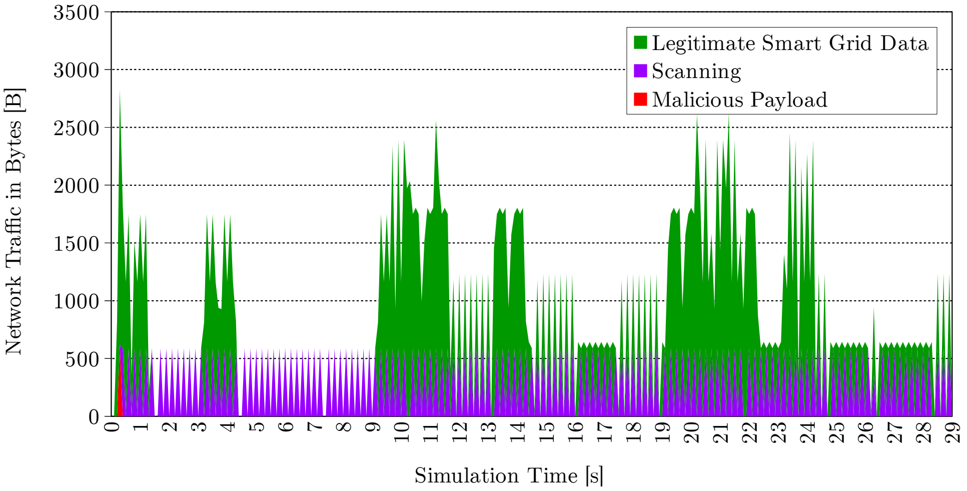


Figure 41: Pandemic malware in centralized topology, pcap example

The extreme infection speed of pandemic malware relies primarily on its small file size and the aggressive scanning strategy. However, we note that there are some time-frames visible in Figure 40 that appear to be idle-times for the infection process, e.g., between 30 and 37 seconds. Furthermore, these idle-times increase in length while time progresses. This is due to aggressive scanning, yielding fewer results with increasing infection ratio, because of the malware is scanning nodes which are already infected. Should the attacker implement an optimized scanning strategy, e.g., hitlist-permutation scanning, cf. Section 4.4.2, the overall infection time could be minimized.

Effective countermeasures for pandemic malware in centralized topologies: We argue that particular attention should be paid to the basic defense measures, cf. Sections 4.3 and 10. They should suffice to stop such an attack. Special attention should be placed on well implemented update policies that patch known vulnerabilities used by this simple malware type, and data integrity checks on inbound data. Since this malware type opens new connections regardless of local traffic, they are easily detectable by IDS. Special attention should be placed on the HL-node's security functions as it represents the single point of failure in this setup.

Endemic Malware

Figures 42 and 43 illustrate endemic malware during the infection phase in a centralized topology, moving slightly slower compared to pandemic malware due to its increased payload size. However, the optimized scanning strategy makes up for the drawbacks a larger payload brings. The scanning strategy used by endemic malware makes aggressive scanning unnecessary and optimizes this traffic accordingly. Additionally, to these network features, a larger payload may enable this malware type with other advanced capabilities, e.g., obfuscation techniques or new exploits, such that endemic malware may be able to overcome basic security measures, cf. Section 4.3. However, we do not simulate these advanced host-based features.

Figure 42 shows that the first ML-node is infected within 9.26 seconds and the HL-node in 10.59 seconds. At this point a destruction attack is possible. The last ML-node is infected within 32.38 seconds after which all ML-nodes could be powered down or spied upon for aggregated data. 75% of all nodes are infected within 43.25 seconds which allows the attacker to spy on selective nodes, or commence other attack types, cf. Section 8.3. All nodes are infected within 51.38 seconds, which is slightly slower compared to pandemic malware. At this point all nodes can be subject to any attack type. However, there are no delays occurring due to the optimizes scanning strategy, leaving only the payload size for optimization.

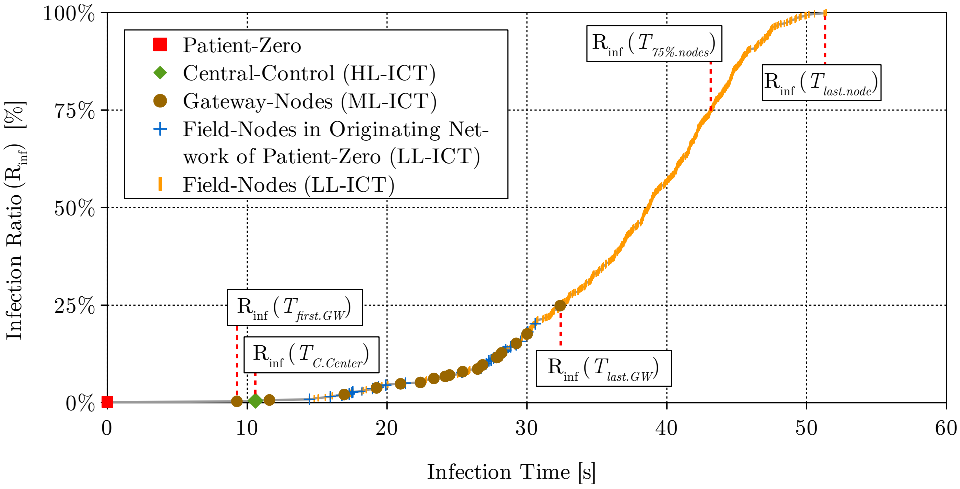


Figure 42: Endemic malware in centralized topology, infection graph

Figure 43 shows legitimate traffic, scanning, and malicious traffic inside one exemplary PLC link. It shows the decreased scanning traffic compared to pandemic malware, and the large payload that must be transferred.

The endemic malware carries more advanced features in its payload, making it the more dangerous adversary compared to pandemic malware when considering that both take about the same time for full infection. However, this case outlines how increased stealthiness at the cost of speed need not impose heavy penalties, yet bring benefit to the attacker at a similar infection ratio.

Effective countermeasures for endemic malware in centralized topologies:

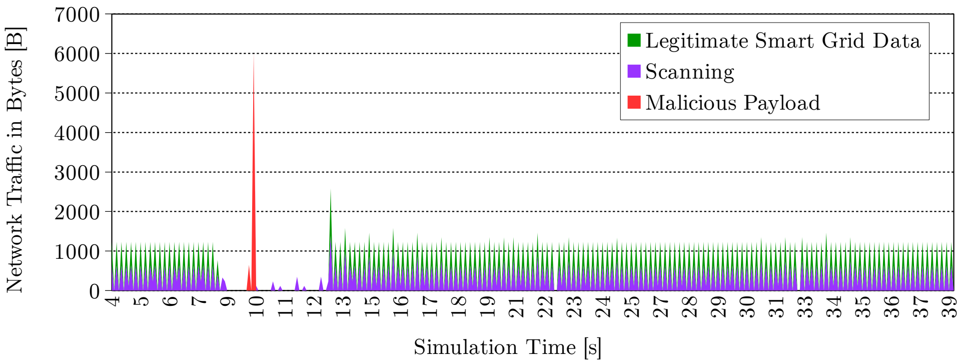


Figure 43: Endemic malware in centralized topology, pcap example

We argue that basic defensive measures, cf. Section 4.3, should represent the bare minimum necessary to have a chance of defeating endemic malware. However, the results in Section 8.1 show that it is unlikely to prevent such an attack without additional defensive measures, as discussed in Section 10. Especially the single point of failure, the HL-node, requires special attention by defenders for it has to manage all controls and the ML-nodes do not act as a protective barrier towards it.

Contagion Malware

Contagion malware is restricted to transfer its payload inside legitimate traffic, making it considerably slower due to waiting periods that occur from the sending cycles finishing. Additionally, it appends the payload on legitimate data flows once transfer has finished. This presents a significant advantage against detection mechanisms, e.g., IDS, because the payload transfer is only visible when examining legitimate traffic, which can be especially challenging when encrypted connections are used, cf. Section 6.9. Figure 44 shows that the first ML-node is infected within 38.63 seconds. This result is due to finishing the remaining legitimate data-sending-cycle, on which the malware can append. The HL-node is infected after 96.91 seconds, thus, about 60 seconds later. The remaining nodes are infected consecutively, whereas two sending cycles occur during the infection phase. The last gateway is infected in 158.43 seconds allowing to power down all ML-nodes. 75% of all nodes are infected within 159.38 seconds and all nodes are infected within 219.54 seconds, enabling all attack types, cf Section 8.3.

The pcap illustration in Figure 45 shows malicious traffic that resides inside legitimate traffic in one exemplary PLC link. We show this example to illustrate that this malware type appends on legitimate traffic without scanning. However, we note that the malicious traffic, in red, does not appear as a

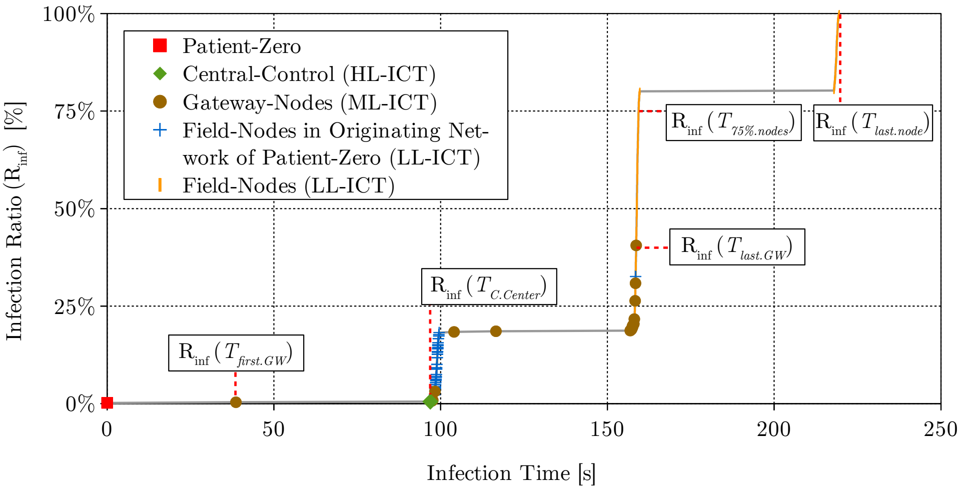


Figure 44: Contagion malware in centralized topology, infection graph

visible separate TCP link.

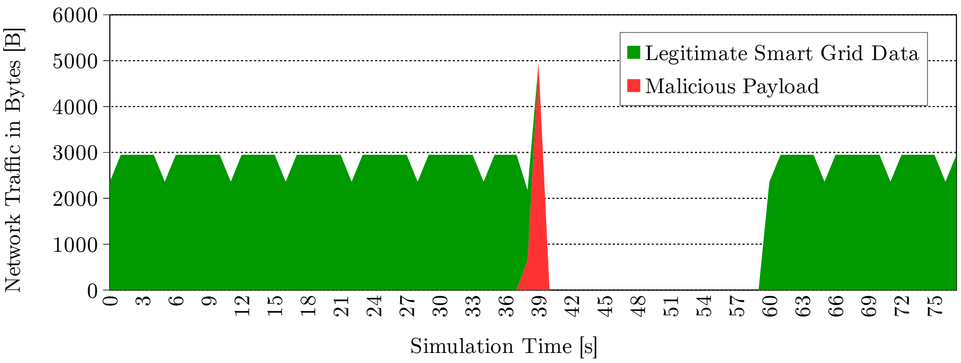


Figure 45: Contagion malware in centralized topology, pcap example

Although contagion malware is slower than the other two malware types, it is the most illusive because it hides inside legitimate connections. Decreased speed makes the payload size less of a problem. However, our current malware traffic could still be detected by its outstanding peak in Figure 45. This leaves room for optimization against detection, but does not increase the infection speed. Additionally, contagion malware carries advanced features inside its large payload making host detection highly unlikely, cf. Section 8.1.

Effective countermeasures for contagion malware in centralized topologies: Effective countermeasures that can defeat such an adversary include, e.g., anomaly detection systems that can investigate TCP traffic for malicious packets but also represent extensive investments, or extensive segmentation with strict policies, and detailed code reviews that prevent such vulnerabilities. Basic features will not suffice against malware that can hide inside legitimate connections and carry advanced anti-detection tools on-board.

Centralized Topology Results Summary

In this section we calculate the efficiency of the three malware types within the centralized topology. For this we use the metrics derived in Section 6. Table 32 summarizes all results for the centralized topology. All malware types manage to infect all nodes in the network, and all in within a short time frame.

Pandemic malware, being the smaller and faster type, has an attack efficiency of 24.43%. Considering that we advise to implement basic defense measures, the noisy scanning and propagation behavior should be easily detected and blocked by automated defenses. However, should pandemic malware manage to break through the basic defense mechanisms, infection can easily outrun any human intervention. The central control node is infected within 1 second, thus, human intervention is not possible. However, standard security measures should prevent it.

Endemic malware has a slightly higher attack efficiency compared to pandemic malware. However, endemic malware is harder to counteract by host-based defenses. Furthermore, the propagation of a larger payload can, in this case be compensated by the more effective scanning strategy. Additionally, endemic malware having greater complexity compared to pandemic malware, may be capable of even more challenging attack vectors that cannot be counteracted by baseline defense measures. Although we can see a clear trade off between speed and stealthiness, endemic malware benefits from that. **Advanced automated defenses, e.g., intrusion detection and prevention may be able to reactively contain this malware type, if implemented.**

Contagion malware follows a hidden channel propagation strategy, resulting in very slow infection. Additionally, this malware type cannot be defected by basic defense mechanisms. Although the infection efficiency is very low, this malware type presupposes an advanced IDS to even have a chance of detecting it. Therefore, contagion malware is raising the bar for defenders, giving it an advantage due to the non-scanning strategy. This also increases its attack-efficiency to 49.39% although full infection takes a long time.

Table 32: Centralized topology - malware efficiency measurements

Metric	Pandemic	Endemic	Contagion
R_{inf} [%]	100.00	100.00	100.00
R_{clean} [%]	0.00	0.00	0.00
$T_{first.GW}$ [s]	0.53	9.26	38.63
$T_{C.Center}$ [s]	0.62	10.59	96.91
$T_{last.GW}$ [s]	18.36	32.38	158.43
$T_{75\%.nodes}$ [s]	18.82	43.25	159.38
$T_{last.node}$ [s]	$T_{all.nodes}$	$T_{all.nodes}$	$T_{all.nodes}$
$T_{all.nodes}$ [s]	49.44	51.38	219.54
$E_{infection}$ [%] [c]	0.645	0.378	0.041
R_{scn} [%]	90.16	98.23	n.a.
R_{uscn} [%]	9.84	1.77	n.a.
E_{scn} [%] [c]	0.39	9.84	n.a.
U_{tr} [%]	2.68	2.87	n.a.
U_{TCP} [%]	3.96	4.89	n.a.
A_{flow} [%]	n.a.	n.a.	1.27
$E_{propag.}$ [%] [c]	96.04	95.00	98.73
$E_{noise.suppress.}$ [%] [c]	48.22	52.42	$E_{propag.}$
E_{attack} [%] [c]	24.43	26.40	49.39
D_{attack} [%] [c]	75.43	73.60	50.61

Notation: (c) Calculated

9.3.2 Cell Topology - Simulation Results

The cell topology is modeled as described in Section 5.4.2 with wired PLC uplinks between LL-nodes and ML-nodes. Some ML-nodes are connected to the HL-node via optical fiber, cf. Section 5.4.2. Additionally, a wireless mesh network exists between the ML-nodes. The ML-nodes in this topology represent intelligent local control nodes with more capabilities compared to the centralized topology. The expected results are fast infection and propagation to the neighboring ML-nodes via the mesh network, that allows bypassing the HL-node. See Table 33 for a summary of the metrics.

Pandemic Malware

Figures 46 and 47 illustrate pandemic malware during the infection phase in a cell topology. Infection of the first ML-nodes commences quickly within 1.86 seconds and spreads to other ML-nodes. The control center is infected after 5.6 seconds, allowing destruction attacks. Should the attacker choose to commence another attack strategy, cf. Section 8.3, the following infection times apply. After 26.87 seconds the attacker has control over all ML-nodes and after 27.67 seconds over 75% of all nodes. At this time the attacker can spy on selected nodes and commence all attack types. After 55.78 seconds the attacker can selectively spy on all nodes, and commence all attack types.

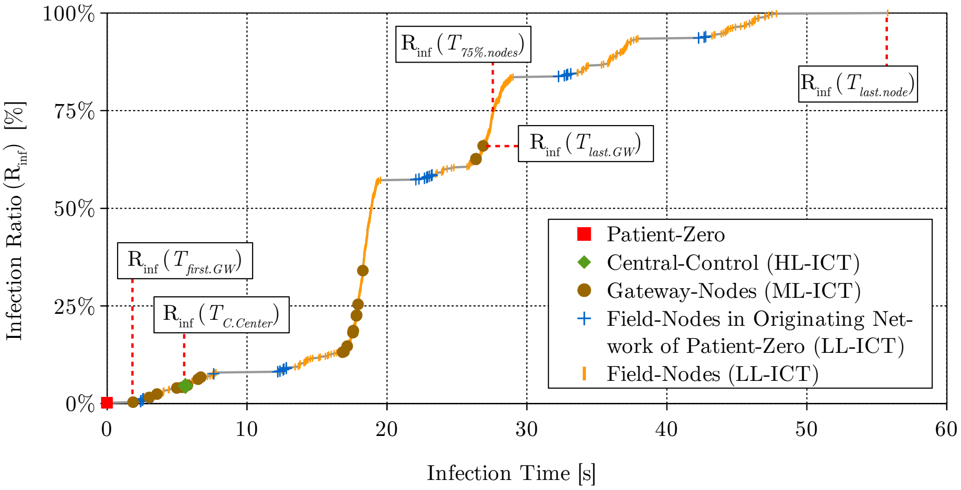


Figure 46: Pandemic malware in cell topology, infection graph

Figure 47 shows legitimate, scanning, and malicious traffic inside the ML-ICT mesh network that exists between the cell controller units. We show this example to illustrate the extensive scanning taking place by pandemic malware. Since the malware scans the entire wireless mesh network its aggressive scanning strategy produces link delays. This results in a large number of redundant scans that exhaust the link bandwidth.

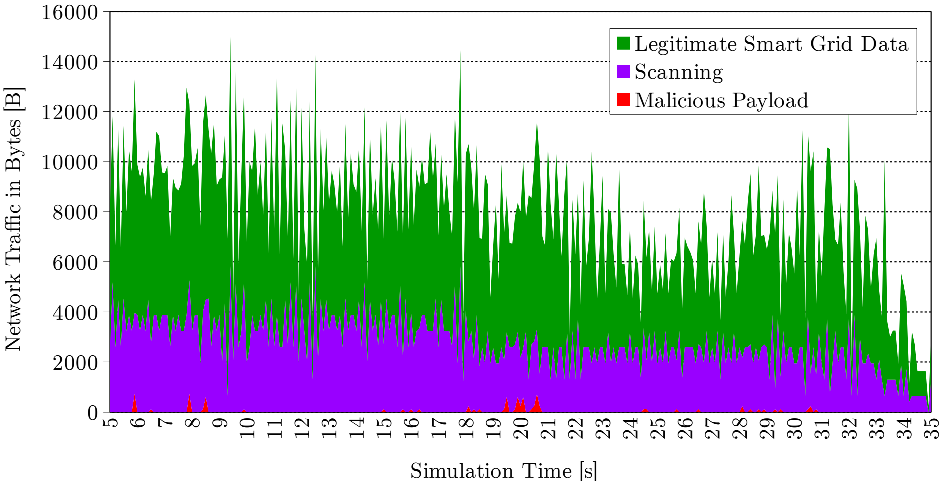


Figure 47: Pandemic malware in cell topology, pcap example

Pandemic malware moves fast inside cell topologies, which are similar to the centralized topology, by comparison. Although slightly slower, malware inside this topology can optimize its spreading, even if some nodes would be immune to the attack, due to alternative routes. Furthermore, the small file size and aggressive scanning strategy of pandemic malware improves the propagation speed. However, it also produces much noise inside the network, saturating the link bandwidth, producing idle-times, e.g., between 9 and 11 seconds. Furthermore, these idle-times do not seem to increase in length while time progresses, as was the case in the centralized topology. We find this is due to the alternative routes available by the mesh network that allows propagation in another direction, should one link be congested. If the attacker implements an optimized scanning strategy, cf. Section 4.4.2, the overall infection time could be reduced even further.

Effective countermeasures for pandemic malware in cell topologies:
We argue that particular attention should be paid to the basic defense measures, cf. Section 4.3. Although they should suffice to stop such an attack, special attention is required for the protection of the ML-nodes because they connect different networks. First, well implemented update policies should stop vulnerabilities that can be exploited by this simple malware type. However, integrity checks, private Virtual Local Area Networks (VLAN's) and white-listing between network segments should also be effective for stopping this malware type from propagation. Pandemic malware opens new connections regardless of local traffic, thus, it is easily detected by IDS. Special focus should be placed on the ML-nodes as the local control nodes.

Endemic Malware

Figures 48 and 49 illustrate endemic malware in a cell topology. It moves slightly slower compared to its pandemic counterpart, which is due to the increased payload size. However, similar to the centralized topology, the optimized scanning strategy makes up for the larger payload, however, with the all the benefits a more complex payload may have. The hitlist-scanning strategy can optimize its noise output according to the reduced search space, thus, is more efficient. Furthermore, a larger payload represents advanced on-board capabilities, e.g., host-based obfuscation and new exploits. These features may allow endemic malware to bypass basic security measures, cf. Section 4.3.

Figure 48 shows that most ML-nodes and the HL-node are infected within a short time, although some LL-nodes are among the early infected nodes. The first ML-node is infected within 4.96 seconds and the HL-node in 10.36 seconds. When the HL-node is compromised a destruction attack is possible. The last ML-node is infected within 32.68 seconds after which all ML-nodes could be powered down. 75% of all nodes are infected within 45.41 seconds which allows the attacker to commence any attack type, cf. Section 8.3, to a significantly large group. All nodes are infected within 52.66 seconds, making endemic malware slightly faster than pandemic malware in cell topologies. At this point all nodes are subject to any attack type.

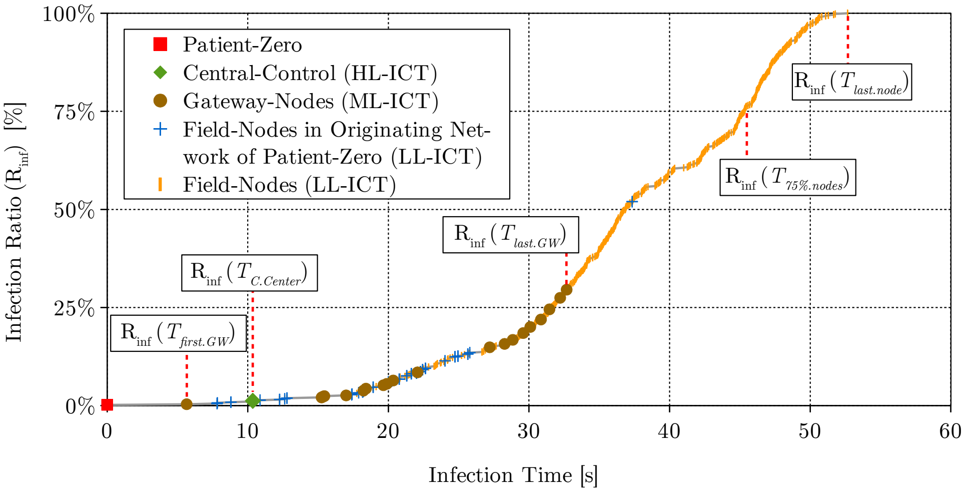


Figure 48: Endemic malware in cell topology, infection graph

Figure 49 shows legitimate traffic, scanning traffic, and malicious traffic in the mesh network. The large payload stands out in traffic, whereas the scanning traffic is less significant, thus, harder to detect.

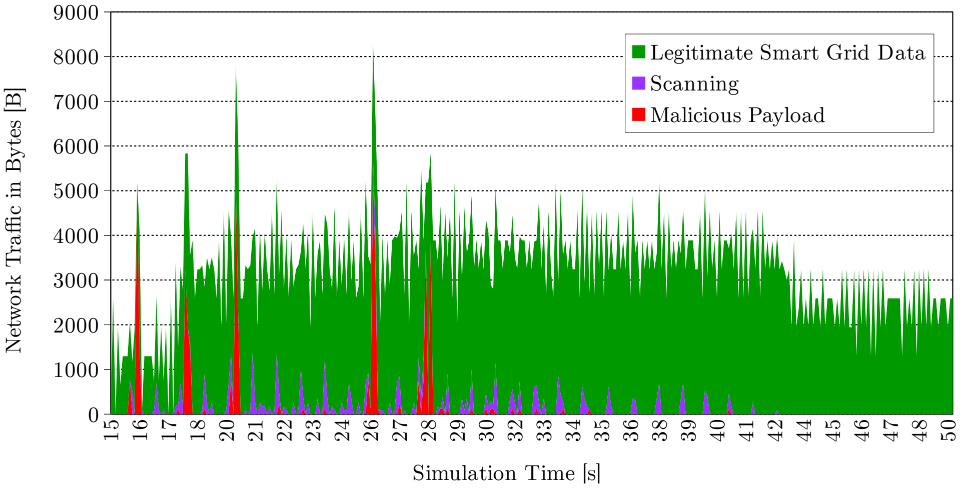


Figure 49: Endemic malware in cell topology, pcap example

Since endemic malware carries more advanced features in its payload, it is the more challenging adversary compared to pandemic malware. When considering that both take about the same time for full infection, this is an advantage. Most of these advanced features, except for the optimized scanning strategy, represent host-based features that make detection of this malware type harder by anti-malware tools.

Effective countermeasures for endemic malware in cell topologies:
We argue that basic defensive measures, cf. Section 4.3, should represent the bare minimum necessary to defeat endemic malware. However, the results in Section 8.1 show that it is unlikely to prevent such an attack without additional defensive measures, as discussed in Section 10. Special attention should be paid to the security of the ML-nodes as they represent the last line of defense before the HL-node can be infected. These neuralgic nodes should be prepared to manage local sub-grids, thus, in the energy domain as well as in the ICT domain. Additionally, the HL-node should be protected as well, and not be neglected.

Contagion Malware

Contagion malware is restricted to transfer its payload inside legitimate traffic, decreasing its propagation speed at the benefit of increased stealthiness, thus, detection-avoidance. It appends its payload on legitimate data once transfer has finished. This gives contagion malware a significant advantage against detection mechanisms because the payload transfer is not visible as unsolicited connections, but instead obfuscated inside legitimate TCP flows. Figure 50 shows that the first ML-node is infected within 79.01 seconds, whose delay is caused by finishing the remaining legitimate data-sending-cycle. The HL-node is infected after 139.86 seconds, allowing the attacker to commence a destruction attack. The last gateway is infected within 201.25 seconds allowing the attacker to power down all ML-nodes. 75% of all nodes are infected within 259.05 seconds and all nodes are infected within 259.77 seconds, enabling all attack types, cf Section 8.3.

Figure 51 illustrates malicious traffic that resides inside legitimate traffic. We show this example to illustrate that this malware type appends on legitimate traffic. However, we note that the malicious traffic, in red, does not appear as a visible separate TCP stream.

Although contagion malware is slower than all other malware types, it moves stealthily due to its hidden propagation capabilities within legitimate connections. Although heavily decreased speed makes the payload size less of a problem, this traffic could still be detected by its outstanding peak, cf. Figure 51, leaving room for optimization. Since contagion malware carries advanced features inside its large payload, it is highly unlikely to be detected even by advanced host-based malware-tools, cf. Section 8.1.

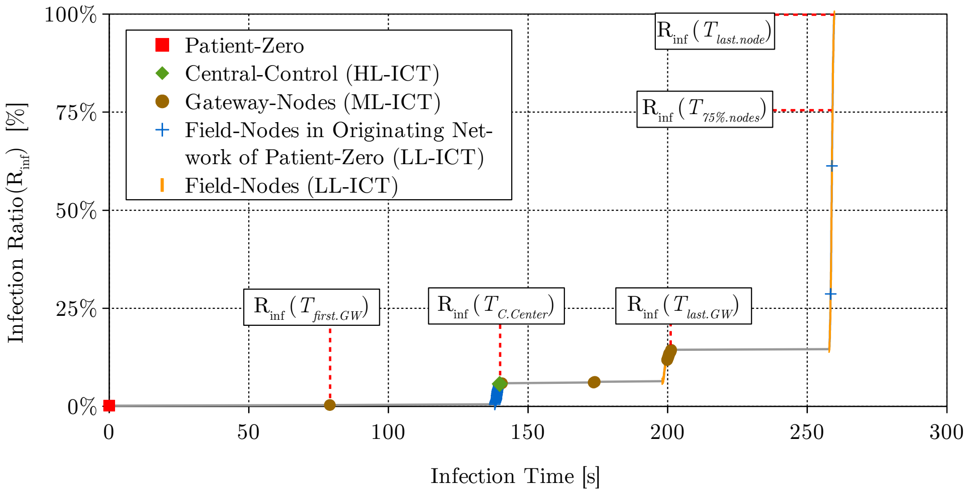


Figure 50: Contagion malware in cell topology, infection graph

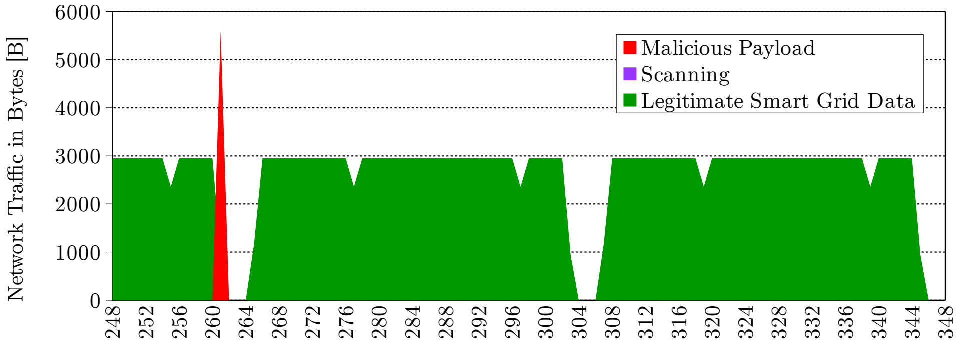


Figure 51: Contagion malware in cell topology, pcap example

Effective countermeasures for contagion malware in cell topologies:
Effective countermeasures that can detect such an adversary include, e.g., anomaly detection systems that can investigate TCP traffic for malicious packets. They may be able to identify the outstanding peak. Basic defense features will not suffice against contagion malware. Special attention should be paid to the ML-nodes as these neuralgic nodes represent the last line of defense before the HL-node can be infected. Additionally, the HL-node should be protected as well, and not be neglected.

Cell Topology Results Summary

We calculate the efficiency of the three malware types within the cell topology, for which we use the metrics derived in Section 6. Table 33 summarizes all results for the cell topology, which provides more options to bypass congested links due to the ML-ICT mesh network.

Pandemic malware, being the smallest and fastest malware type, shows high propagation efficiency due to its small payload size, low infection efficiency due to its propagation speed, and low scanning efficiency due to its aggressive scanning strategy. Therefore, it should be easy to defend against by baseline measures.

Endemic malware has an increased attack efficiency which results from the more advanced scanning behavior inside the network, that also optimizes propagation and infection. Although the larger payload slows this malware type, other factors manage to optimize its infection and propagation to compensate drawback compared to pandemic malware.

Contagion malware follows a hidden propagation strategy, resulting in slow infection. However, its increased propagation efficiency makes up for its slow infection. Furthermore, this malware type cannot be defected by basic defense mechanisms. Contagion malware presupposes advanced IDS to even have a chance of detecting it.

Table 33: Cell topology - malware efficiency measurements

Metric	Pandemic	Endemic	Contagion
R_{inf} [%]	100.00	100.00	100.00
R_{clean} [%]	0.00	0.00	0.00
$T_{first.GW}$ [s]	1.86	4.96	79.01
$T_{C.Center}$ [s]	5.60	10.36	139.86
$T_{last.GW}$ [s]	26.87	32.68	201.25
$T_{75\%.nodes}$ [s]	27.67	45.41	259.05
$T_{last.node}$ [s]	$T_{all.nodes}$	$T_{all.nodes}$	$T_{all.nodes}$
$T_{all.nodes}$ [s]	55.78	52.66	259.77
$E_{infection}$ [%] [c]	0.078	0.421	0.031
R_{scn} [%]	79.52	98.10	n.a.
R_{uscn} [%]	20.48	1.90	n.a.
E_{scn} [%] [c]	0.29	9.08	n.a.
U_{tr} [%]	0.39	1.33	n.a.
U_{TCP} [%]	19.04	15.99	n.a.
A_{flow} [%]	n.a.	n.a.	9.28
$E_{propag.}$ [%] [c]	80.96	84.01	90.72
$E_{noise.suppress.}$ [%] [c]	40.63	46.55	$E_{propag.}$
E_{attack} [%] [c]	20.35	23.48	45.38
D_{attack} [%] [c]	79.65	76.52	54.62

Notation: (c) Calculated

9.3.3 Mesh Topology - Simulation Results

The mesh topology, as introduced in Section 5.4.3, manages 9 mesh networks and one controlling node each, represented by 8 ML-nodes and 1 HL-node. We decided to split the entire topology into several sub-mesh networks because of the limitations which are discussed in Section, 9.1.2. Each ML-node is connected to the HL-node by a P2P link to upload the collected data and receive commands. Although we expect benefits for the network traffic by the mesh networks, we suspect that malware can utilize the same benefits. Table 34 provides a summary of the metrics.

Pandemic Malware

Figures 52 and 53 illustrate pandemic malware in a mesh topology. The infection of the first ML-node commences after most of the local LL-nodes are infected (34.8s). Immediately afterwards the HL-node is infected (35.38s), which in turn infects all remaining ML-nodes (36s). At this point destruction attacks can commence against the infiltrated backhaul network. After 65.6 seconds 75% of all nodes are infected which allows the attacker to spy on selected nodes and commence all attack types, cf. Section 8.3. Only after 350.3 seconds the attacker can selectively attack all nodes. This result shows that 75% of all nodes represents a reasonably large group which can be attacked within a short time frame compared to how long it takes to infect all nodes.

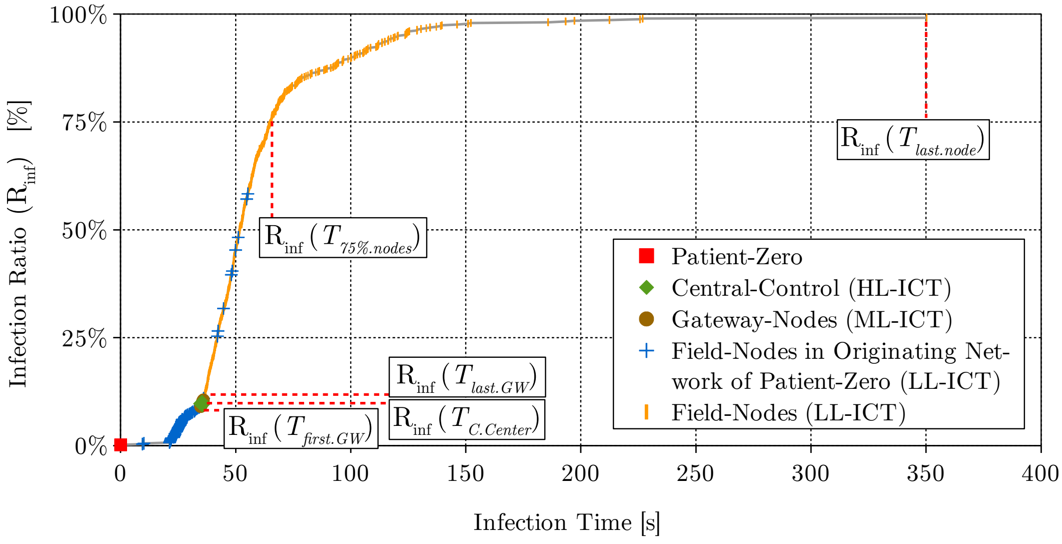


Figure 52: Pandemic malware in mesh topology, infection graph, presented in [46]

Figure 53 shows legitimate, scanning, and malicious traffic in one mesh network. We illustrate the extensive scanning of pandemic malware, producing link delays that are exhausting the link bandwidth. Figure 53 shows decreasing scanning traffic which is due to the exhaustion of the search space.

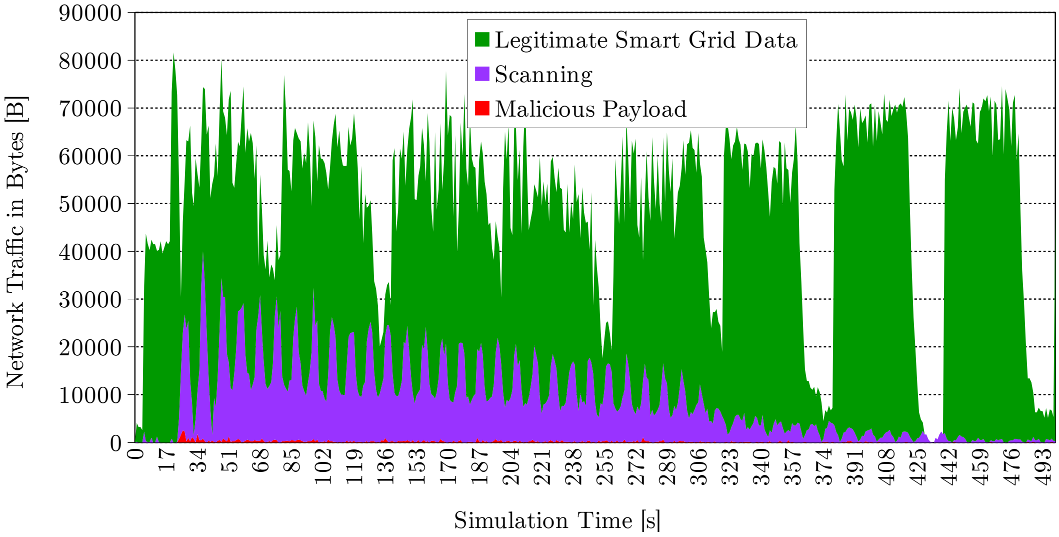


Figure 53: Contagion malware in mesh topology, pcap example, presented in [46]

Pandemic malware manages to infect the ML-nodes and HL-node within a short period of time, allowing to quickly commence destruction attacks. However, the results also show that 75% of all nodes, thus, the majority, are infected shortly after, allowing to commence the remaining attack types. The long period required to infect the remaining nodes hardly seems worthwhile from an attackers point of view. These results show that when comparing the same malware type with other topologies, the difference is significantly smaller compare to the same points for full infection. We find that the attacker could implement an optimized scanning strategy to reach the backhaul nodes even more quickly, however, this example already represents a significant challenge for defenders due to the spreading speed.

Effective countermeasures for pandemic malware in mesh topologies:
We argue that particular attention should be paid to the basic defense measures, cf. Section 4.3. They should suffice to stop such a highly visible attack due to the aggressive scanning strategy. Special attention is required for the protection of the ML-nodes, providing a direct path to the HL-node. A progressive update policy should prevent vulnerabilities, and integrity checks, VLAN's, and white-listing between network segments should also be effective measures.

Endemic Malware

Figures 54 and 55 illustrate endemic malware in a mesh topology. Compared to pandemic malware we notice a slight advantage in infection speed, which is due to the optimized scanning, despite the large payload. Figure 54 shows that the first ML-node is infected within 21.74 seconds. Although some LL-nodes are among the early infected nodes, the HL-node is infected within 26.84 seconds. At this point a destruction attack against the entire network is possible. The last ML-node is infected within 36.61 seconds after which all ML-nodes can be powered down. 75% of all nodes are infected within 73.17 seconds which allows the attacker to commence any attack type to a significantly large group. The last node is infected within 153.45 seconds, however, in this case, full infection is not accomplished. This does not matter though, because 75 % of all nodes are being infected, representing a large enough set of nodes that a given target is likely to be among them.

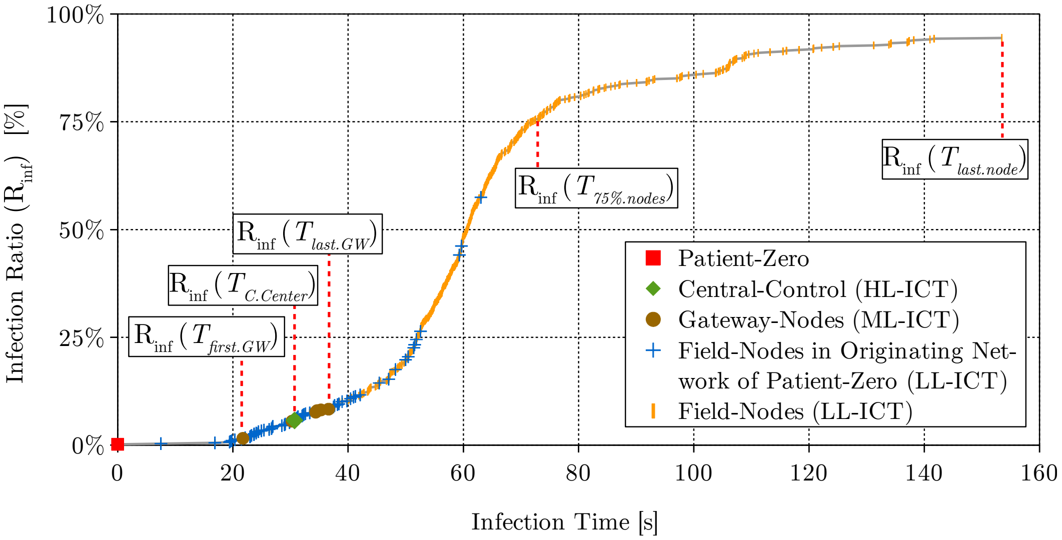


Figure 54: Endemic malware in mesh topology, infection graph, presented in [46]

Figure 55 illustrates legitimate, scanning, and malicious traffic inside a mesh network. The large payload stands out compared to the scanning traffic which is reduced.

The endemic malware carries more advanced features, making it the more challenging adversary compared to pandemic malware, taking into account that both require about the same time to infect the last gateway.

Effective countermeasures for endemic malware in cell topologies:
We argue that basic defensive measures, cf. Section 4.3, only represent the

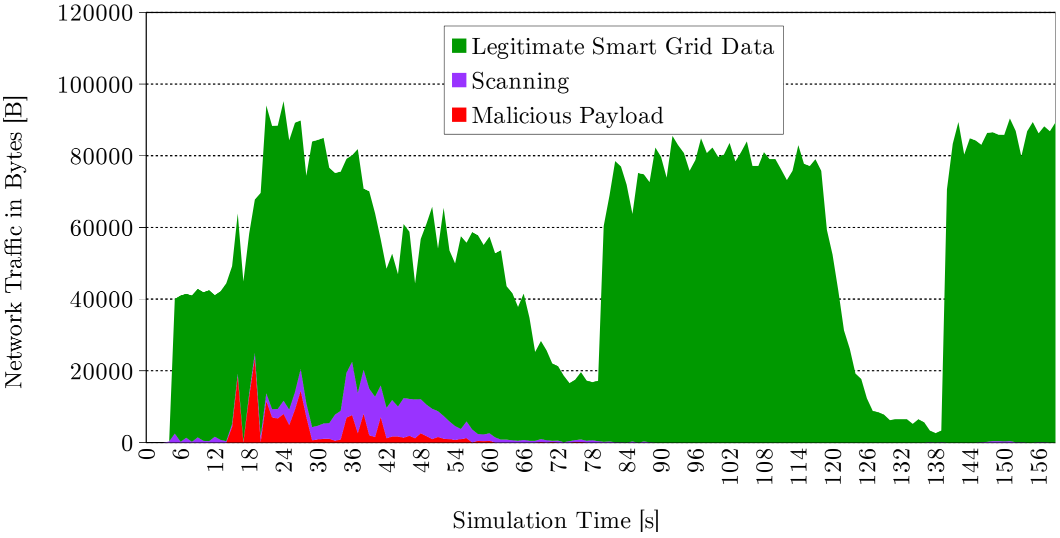


Figure 55: Contagion malware in mesh topology, pcap example, presented in [46]

bare minimum necessary to defeat endemic malware. However, the results in Section 8.1 show that it is unlikely to prevent such an attack without additional defensive measures, as discussed in Section 10. Special attention should be paid to the ML-nodes as they represent the entry points to the backhaul network, thus, to the HL-node.

Contagion Malware

Contagion malware is restricted to transfer its payload inside legitimate traffic, decreasing the propagation speed due to fixed sending cycles. Contagion malware has a significant advantage against detection mechanisms because the payload transfer is only visible as a peak among legitimate TCP connections, cf. Section 6.9. Figure 56 shows that the first ML-node is infected within 68.65 seconds. The HL-node is infected after 106.66 seconds and the last gateway is infected within 154 seconds allowing the attacker to power down all ML-nodes. 75% of all nodes are infected within 972.08 seconds and the last nodes is infected within 2298.00 seconds, enabling all attack types, cf Section 8.3. Figure 51 shows malicious traffic residing inside legitimate traffic. We show this example to illustrate that this malware type appends on legitimate traffic. However, we note that the malicious traffic does not appear as a visible separate TCP stream.

Although contagion malware is slow, it moves stealthily within legitimate connections. Heavily decreased speed makes the large payload size less of a

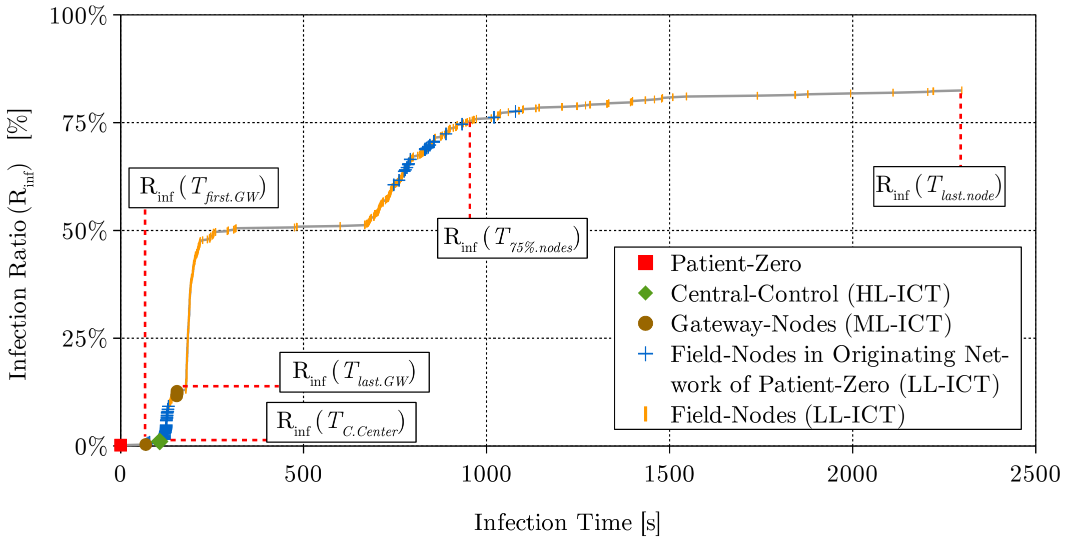


Figure 56: Contagion malware in mesh topology, infection graph, presented in [46]

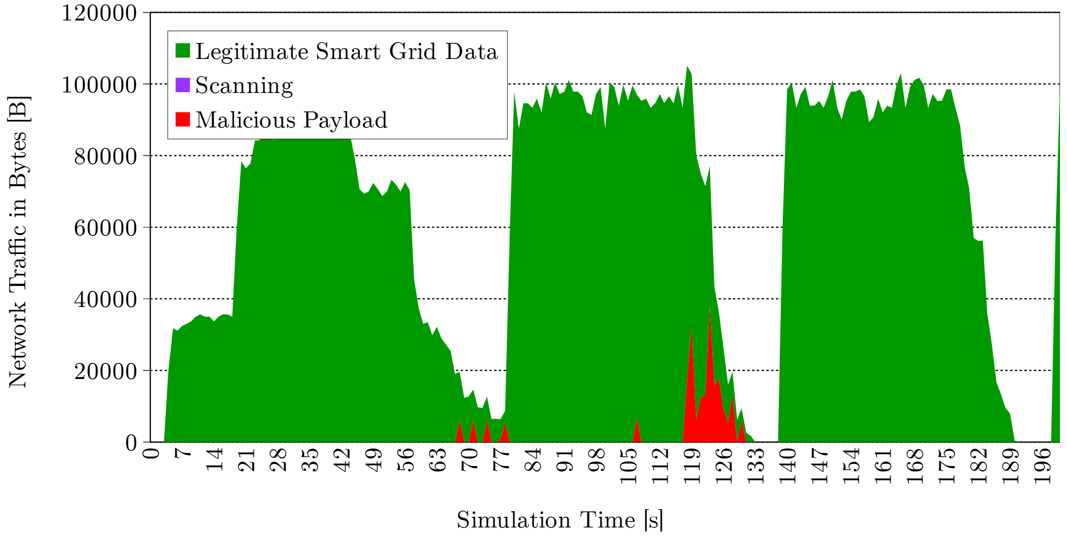


Figure 57: Contagion malware in mesh topology, pcap example, presented in [46]

problem and the malicious traffic could still be detected if traffic is monitored closely and correlated against legitimate traffic.

Effective countermeasures for contagion malware in cell topologies:
Effective countermeasures that can detect such an adversary include, e.g., anomaly detection systems that can investigate TCP traffic for malicious

packets. Pattern based detection methods may identify the malicious payload by suspicious data flow inside a TCP connection. However, traffic based intrusion detection would already fail in detecting this malware type. Basic features will not suffice against contagion malware.

Mesh Topology Results Summary

Table 34 summarizes all results for the mesh topology. Pandemic malware, has an attack efficiency of 17.25%, represented by a medium propagation efficiency, a low infection efficiency, and low scanning efficiency. Therefore, pandemic malware should be easy to defend against, when basic measures are implemented.

Endemic malware has in this case a slightly decreased attack efficiency compared to the pandemic malware. This is in part due to the large payload leaving a greater footprint in transit. However, the infection efficiency and scanning efficiency are significantly higher, compensating for the drawbacks of a larger payload. Overall these factors manage to optimize the infection and propagation to regain benefit compared to pandemic malware.

Contagion malware follows a hidden propagation strategy, resulting in slow infection. However, its increased propagation efficiency makes up for the slow infection speed. The nearly perfect propagation efficiency is due to the extensive obfuscation capabilities, which can only be detected when examining legitimate traffic. Furthermore, this malware type cannot be defeated by basic defense mechanisms. Contagion malware presupposes advanced IDS to even have a chance of detecting it.

Table 34: Mesh topology - malware efficiency measurements

Metric	Pandemic	Endemic	Contagion
R_{inf} [%]	99.10	94.44	82.47
R_{clean} [%]	0.9	5.56	17.54
$T_{first.GW}$ [s]	34.80	21.74	68.65
$T_{C.Center}$ [s]	35.38	26.84	106.66
$T_{last.GW}$ [s]	36.00	36.61	154.00
$T_{75\%.nodes}$ [s]	65.60	73.17	972.08
$T_{last.node}$ [s]	350.00	153.45	2298.08
$T_{all.nodes}$ [s]	∞	∞	∞
$E_{infection}$ [%] ^{c}	0.004	0.054	0.014
R_{scn} [%]	98.80	82.65	n.a.
R_{uscn} [%]	1.20	17.35	n.a
E_{scn} [%] ^{c}	0.10	1.13	n.a.
U_{tr} [%]	0.48	3.85	n.a.
U_{TCP} [%]	31.10	32.98	n.a.
A_{flow} [%]	n.a.	n.a.	4.66
$E_{propag.}$ [%] ^{c}	68.90	67.02	95.34
$E_{noise.suppress.}$ [%] ^{c}	34.50	34.07	$E_{propag.}$
E_{attack} [%] ^{c}	17.25	17.06	47.68
D_{attack} [%] ^{c}	82.75	82.94	52.32

Notation: (c) Calculated

9.3.4 Decentralized Topology - Simulation Results

The decentralized topology is modeled as introduced in Section 5.4.4 with sub-meshes that are connected via a backhaul infrastructure. However, there is no central node available, but instead all ML-nodes are managed equally. Furthermore, the infected nodes can reach other nodes in neighboring mesh networks via a layer 2 [82] vulnerability, as per the vulnerability abstraction in Section 4.6. This allows them to jump across networks without using the backhaul infrastructure. With this setup we can achieve similar behavior as if all nodes were in one single mesh network.

Pandemic Malware

Figure 58 illustrates the results of pandemic malware during the infection phase. The infection commences similar to the mesh topology. However, the ML-nodes are infected much slower because they are no longer required for the infection of the other nodes and the malware favors to move laterally without using the backhaul infrastructure. Therefore, ML-nodes are infected as a side effect rather than a stepping stone to other networks. No priority is given to infecting the ML-nodes over LL-nodes.

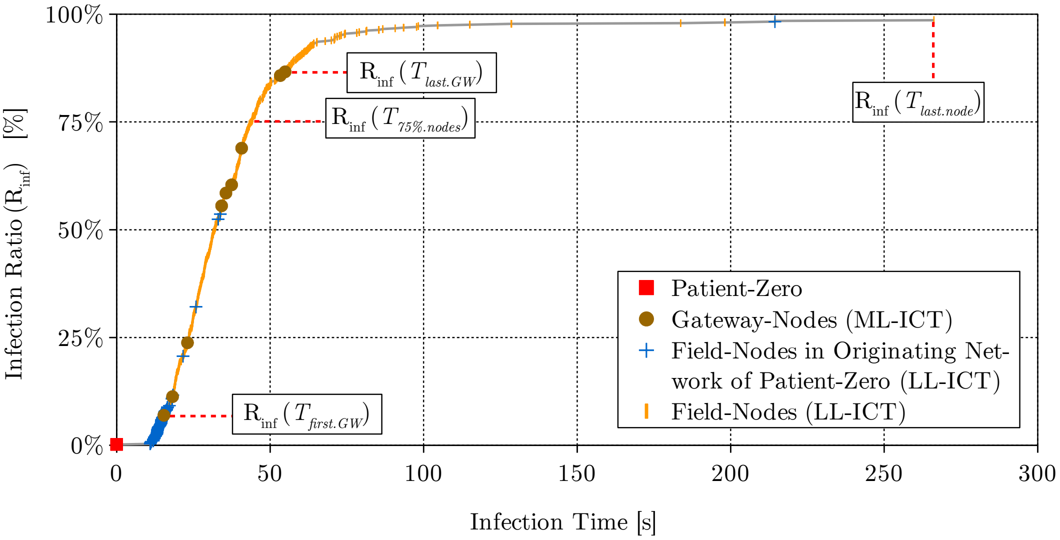


Figure 58: Pandemic malware in decentralized topology, infection graph

Figure 59 shows legitimate, scanning, and malicious traffic in one exemplary mesh network. We show this example to illustrate the extensive scanning taking place by pandemic malware. Because the entire wireless mesh network is scanned, the aggressive scanning strategy produces link delays resulting in a large number of redundant scans that are exhausting the link bandwidth. The illustration shows decreasing scanning traffic which is due to the exhaustion of the search space.

Pandemic malware infects the ML-nodes as a side effect while propagating through the mesh networks. The first ML-node is infected after 15.04 seconds, followed by several LL- and ML-nodes. The 75% infected mark is reached at 43.44 seconds, before the last ML-node is infected after 54.89 seconds. At his point all attack types can commence against this network. However, we note that this simulation shows a long tail, with the last node being infected only after 266.19 seconds. We find that the scanning and propagation strategy of pandemic malware is not optimized for this topology type. Yet, a destruction attack can only commence after 75% of all nodes have been infected. This is noteworthy because at this point all other attack types may also commence.

Effective countermeasures for pandemic malware in mesh topologies:
We argue that particular attention should be paid to the basic defense measures, cf. Section 4.3. They should suffice to stop such a highly visible attack. No special attention is required for the protection of the ML-nodes because they are infected as a side effect rather than targeted, relieving pressure

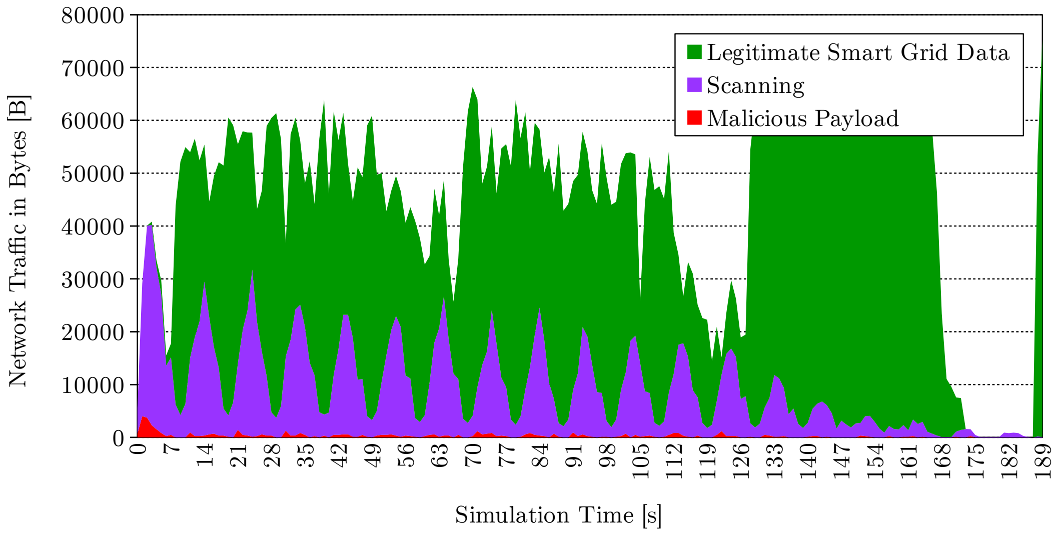


Figure 59: Pandemic malware in decentralized topology, pcap example

off critical nodes. Therefore, progressive updates should prevent vulnerabilities, expected to be exploited by this simple malware type. Furthermore, the previously effective measures, i.e., integrity checks, VLAN's and whitelisting, may not be as effective as expected because the malware can jump to neighboring networks. Regardless of the level of effectiveness, we recommend host-based IDS that places focus on the individual protection of field nodes.

Endemic Malware

Figures 60 and 61 illustrate endemic malware in a mesh topology. Compared to pandemic malware we notice a significant decrease in infection speed. This is due to the increased payload size. Although an optimized scanning strategy may help improving some of the speed drawbacks, it still does not catch up to the benefits aggressive behavior brings with a small payload in this topology. However, the large payload represents advanced on-board capabilities, e.g., host-based obfuscation or new exploits, which can aid endemic malware in obfuscating from hose-based detection measures, cf. Section 4.3. Figure 60 shows that the first ML-node is infected within 20.64 seconds. Then, most of the LL-nodes and ML-nodes are infected successively. 75% of all nodes are infected after 118.59 seconds and the last gateway is infected after 132.52 seconds which allows commencing all attack types, cf. Section 8.3. Shortly after, at 177.42 seconds, the last node is infected.

Figure 61 shows legitimate, scanning, and, malicious traffic inside a mesh network. The payload stands out compared to the scanning traffic.

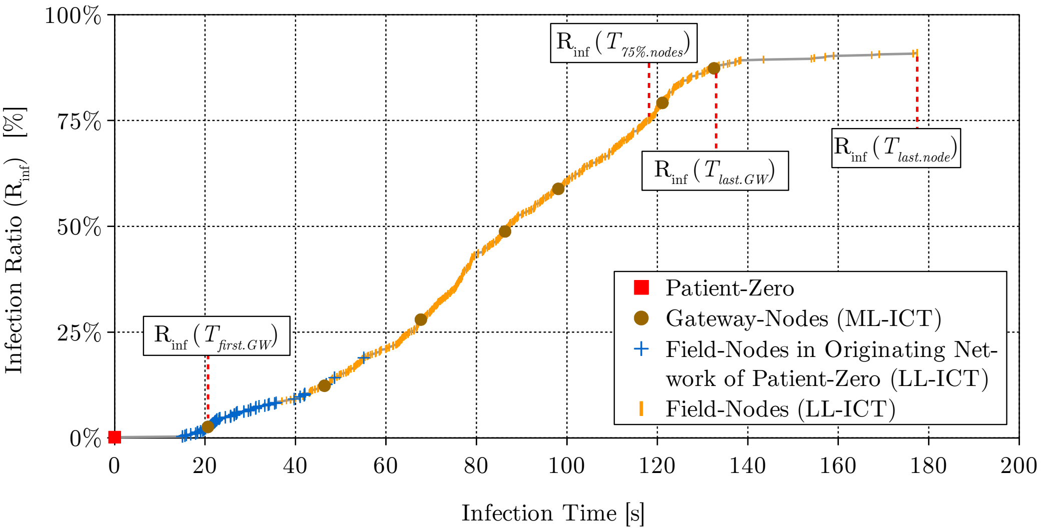


Figure 60: Endemic malware in decentralized topology, infection graph

Endemic malware carries advanced features in its payload which, except for the optimized scanning strategy, represent host-based features that make detection of this malware type harder by anti-malware tools.

Effective countermeasures for endemic malware in cell topologies:
We argue that basic defensive measures, cf. Section 4.3, represent the minimum level of defenses to defeat this malware type, although the results in Section 8.1 show that it is unlikely to prevent such an attack without additional defense measures, as discussed in Section 10. Focus should be placed on preventative measures because there are no neuralgic nodes in this setup. Therefore, the infection can spread regardless of the backhaul network, rendering ML-nodes that shut down their interfaces to prevent further propagation useless.

Contagion Malware

Contagion malware transfers its payload inside legitimate traffic, decreasing the propagation speed. Contagion malware has a significant advantage against detection mechanisms because the payload transfer is only visible among legitimate TCP connections. Figure 62 shows that the first ML-node is infected within 63.34 seconds. The last gateway is infected within 148.29 seconds allowing the attacker to power down all ML-nodes. 75% of all nodes are infected within 1216.85 seconds, including a waiting period which occurs due to the sending cycles of legitimate traffic. At this point all attack types may commence, cf Section 8.3. The last nodes is infected within 2293.29 seconds. Because contagion malware will only append on legitimate TCP

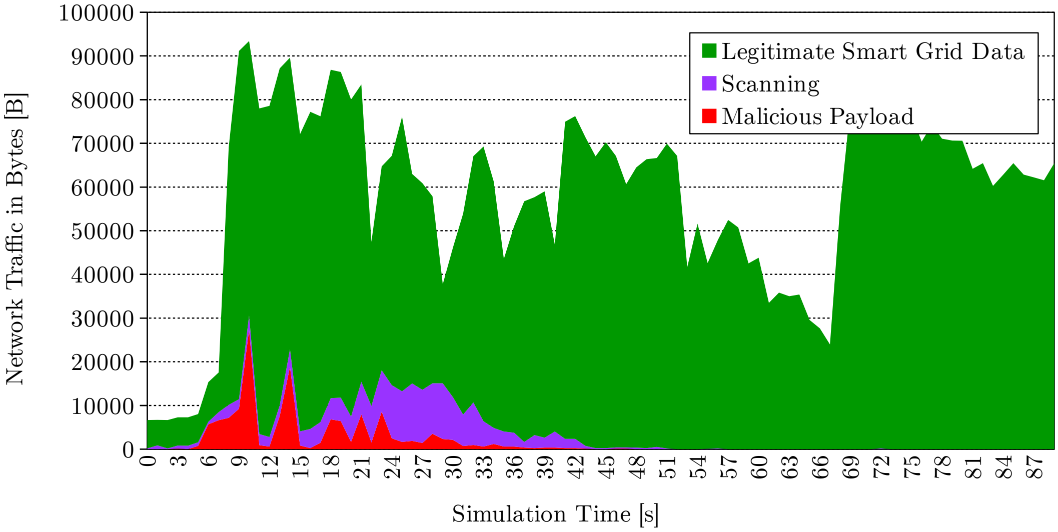


Figure 61: Endemic malware in decentralized topology, pcap example

connections, it cannot jump across sub-meshes the same way pandemic and endemic malware can. However, contagion malware can still move hidden over the backhaul infrastructure, although, this takes longer. Therefore, contagion malware has no significant advantage in decentralized topologies.

Figure 63 shows malicious traffic inside legitimate traffic. We show this example to illustrate that this malware type appends on legitimate traffic without scanning. We note that the malicious traffic does not appear as a visible separate TCP stream.

Although contagion malware is slow compared to other types, it moves hidden within legitimate connections. Heavily decreased speed makes the large payload size less of a problem, because this malware type has increased stealthiness. The malicious traffic could still be detected if traffic is monitored closely, cf. Figure 51. However, defenders must investigate legitimate TCP connections and correlate legitimate to malicious flows. Since contagion malware carries advanced features inside its payload, it is highly unlikely to be detected even by advanced host-based detection systems, cf. Section 8.1, especially when they are connection based.

Effective countermeasures for contagion malware in cell topologies:
Effective countermeasures that can detect such an adversary include, e.g., anomaly detection systems that investigate traffic for malicious packets. Pattern based detection methods may identify the malicious payload by suspicious data flow. Basic defensive features will not suffice against contagion malware. This malware type can be stopped in this network setup by a well equipped ML-node that denies it access to the backhaul network because

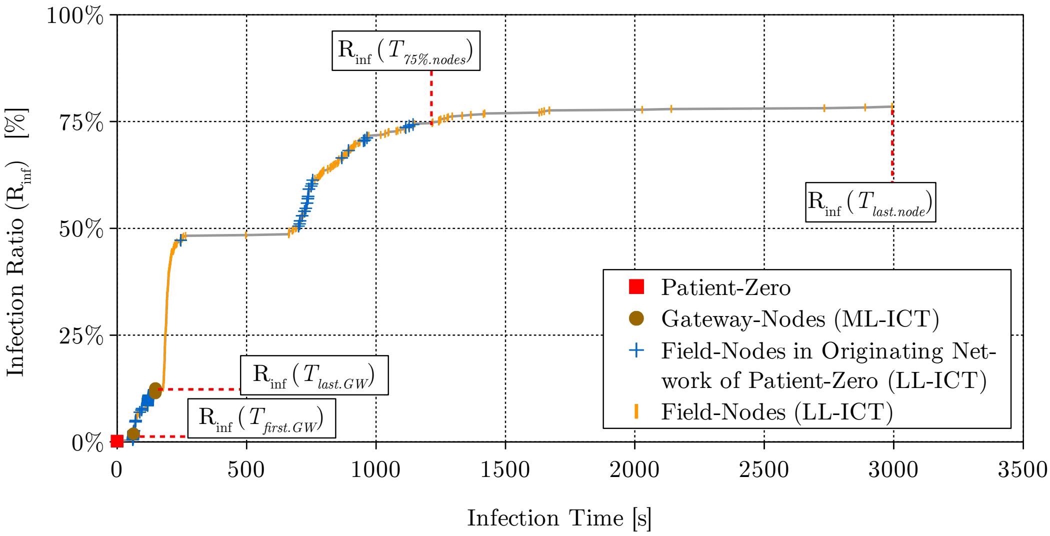


Figure 62: Contagion malware in decentralized topology, infection graph

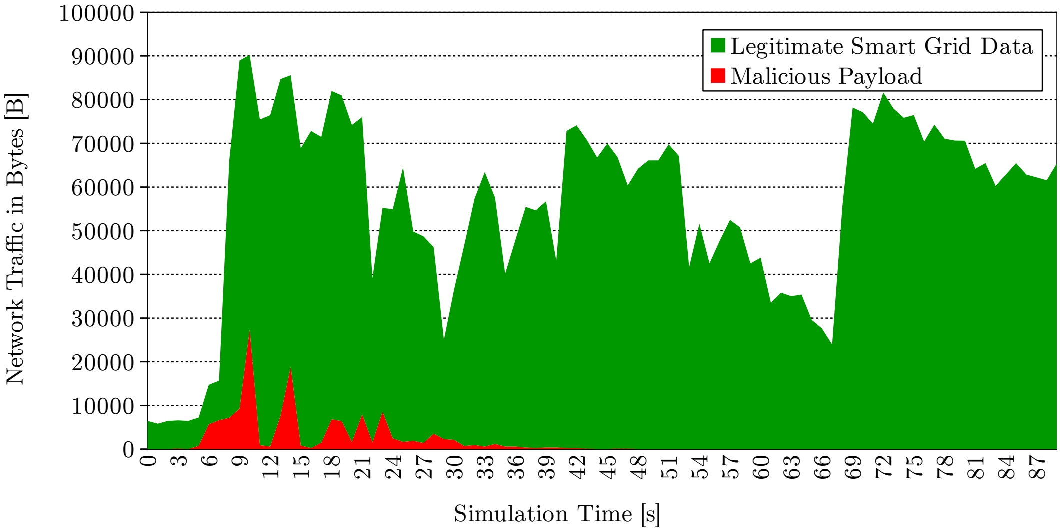


Figure 63: Pandemic malware in decentralized topology, pcap example

contagion malware does not open unsolicited TCP connections and has to travel over the backhaul network for propagation.

Decentralized Topology Results Summary

Table 35 summarizes all results for the decentralized topology. Pandemic malware has the lowest attack efficiency of 15.04%, represented by a medium propagation efficiency, a very low infection efficiency, and a very low scanning efficiency, caused by the aggressive scanning strategy. Therefore, pandemic malware should be easy to defend against when basic measures are implemented.

Endemic malware shows better attack efficiency compared to the pandemic malware in this topology. This is partly due to the slightly increased propagation efficiency, and the much higher scanning efficiency. The infection efficiency is in both cases very low.

Contagion malware follows a hidden propagation strategy, resulting in slow infection. However, its increased propagation efficiency of 97.71% makes up for the slow infection speed, especially because of its increased stealthiness. The extensive obfuscation capabilities, which can only be detected when examining legitimate traffic also include host-based features that may stand fast against a myriad of defense measures. Moreover, contagion malware presupposes advanced IDS to even have a chance of detecting it.

Table 35: Decentralized topology - malware efficiency measurements

Metric	Pandemic	Endemic	Contagion
R_{inf} [%]	84.63	90.97	78.47
R_{clean} [%]	15.63	9.03	21.53
$T_{first.GW}$ [s]	15.04	20.64	63.34
$T_{C.Center}$ [s]	n.a.	n.a.	n.a.
$T_{last.GW}$ [s]	54.89	132.52	148.29
$T_{75\%.nodes}$ [s]	43.44	117.45	1216.85
$T_{last.node}$ [s]	266.19	177.42	2993.29
$T_{all.nodes}$ [s]	∞	∞	∞
$E_{infection}$ [%] c	0.03	0.012	0.010
R_{scn} [%]	81.63	35.73	n.a.
R_{uscn} [%]	18.37	64.27	n.a.
E_{scn} [%] c	1.07	11.12	n.a.
U_{tr} [%]	1.04	4.16	n.a.
U_{TCP} [%]	37.41	21.78	n.a.
A_{flow} [%]	n.a.	n.a.	2.29
$E_{propag.}$ [%] c	62.59	78.22	97.71
$E_{noise.suppress.}$ [%] c	31.83	39.64	$E_{propag.}$
E_{attack} [%] c	15.92	19.83	48.86
D_{attack} [%] c	84.08	80.17	51.14

Notation: (c) Calculated

Table 36: Summary of the Simulation Results

Metric [%]	Pandemic	Endemic	Contagion
Centralized E_{attack}	24.43	26.40	49.39
Centralized D_{attack}	75.43	73.60	50.61
Cell E_{attack}	20.35	23.48	45.38
Cell D_{attack}	79.65	76.52	54.62
Mesh E_{attack}	17.25	17.06	47.68
Mesh D_{attack}	82.75	82.94	52.32
Decentralized E_{attack}	15.92	19.83	48.86
Decentralized D_{attack}	84.08	80.17	51.14

9.3.5 Summary of the Simulation Results

This section summarizes the attack efficiency and defendability results for all topologies and malware types.

We discovered that as soon as mesh networks are used in the topology, the simple topological-scanning strategy (used by pandemic malware) becomes a major drawback because its noise output enables defenders to easily discover it on ML-nodes that may be listening on the network with IDS functionalities. The reason being mesh networks allow ML-nodes to identify misbehaving LL-nodes, which is more difficult to accomplish in P2P-link-based PLC networks. However, the optimized hitlist-scanning strategy (used by endemic malware) holds an advantage over topological-scanning in small mesh networks because of the small number of nodes which are scanned quickly.

When comparing both malware types (pandemic and endemic) which are scanning the network with the silent contagion malware, the major benefit of stealthiness over speed becomes apparent, thus, confirming our discoveries in Section 8.2. Additionally, the advantage of stealthiness over speed remains constant throughout all topologies, making the contagion malware the most effective attack malware in our simulation environment. Pandemic malware and endemic malware decrease in attack efficiency when mesh networks increase in size, however, contagion malware does not, but instead maintains its high level.

Furthermore, the *centralized topology* provides two links (1 PLC and 1 LWL) until the central control node can be infected, cf. Table 19. Although the PLC link is very slow, this topology can generally be infected the fastest, should defensive measures fail to protect the central control node. This is also the most significant drawback of the centralized topology, i.e., high vulnerability against malware propagation, at the highest level of remote control over all subjacent nodes. This includes the ML-nodes which in this case have no advanced control function.

The *cell topology* presents an additional layer of security with its more intelligent ML-nodes. Should the defensive measures at this level fail, as is the case in our simulations, the infection and propagation behavior is similar to the centralized topology, i.e., very quick and devastating to defenders. This result underlines the importance of functioning defensive measures on the controlling neuralgic field nodes (ML-nodes) and how important they are for the rest of the networks stability.

The *mesh topology* replaces the cells from the cell topology with mesh networks that are interconnected via a backhaul infrastructure. This setup too has ML-nodes that manage their local mesh network. However, mesh networks provide one advantage over P2P links, namely, that ML-nodes can listen into the network and identify misbehaving LL-nodes that scan or infect other nodes. They could then counteract this behavior. However, should defensive measures fail, the mesh networks provide the same benefits of highly resilient connectivity that make them more resilient to failure, to the malware as well. Malware can propagate more reliably in mesh networks, whereas compared to P2P links in the previous topologies, shutting these neuralgic nodes down suffices to halt malware propagation in one particular network.

The *decentralized topology* interconnects all mesh networks into one large mesh network containing all nodes. Malware can rampage freely in the case of dysfunctional defensive measures. However, scanning and propagation behavior, although difficult to contain in a highly resilient mesh network, are easily detected by ML-nodes and the HL-node that can operate IDS and anomaly detection systems. Moreover, these functions may not help much against malware propagation because interconnected networks allow many alternative propagation paths.

10 Defense Measures: Results and Discussion

Notice of adoption from previous publications:
Parts of the contents of this chapter have been previously published in [43] discussing defensive measures that build upon the available attack types. [43] includes defensive measures utility companies can implement to increase their resilience against cyber-attacks. The author developed this list of measures continually over the course of this work, whereas the co-author contributed usability and feasibility improvements. This list was also presented in [41].

It is noteworthy that the topology type plays a role in terms of overall system defendability [44, 183]. This is particularly true for critical infrastructures such as power grids and future smart grid implementations that utilize communication networks. Without any measures beyond baseline measures, however, secure architectures will not suffice to repel attackers, cf. Section 4.3. This section complements the previously established security model of our selected defense baseline with additional measures. Our results, cf. Table 37 were published in [41, 43], and correspond to research question 4, cf. Section 1.3, which considers the overall attack-defendability of a network topology. We elaborate on effective measures for smart grid communication networks and arrange them by two categories, namely, field of efficacy and threat level, which can support grid operators in improving their network security. The list is based on the following sources: [2, 8, 11, 20, 26, 28, 44, 49, 59, 66, 69, 71, 80, 81, 84, 104, 113, 114, 122, 124, 125, 127, 144, 152, 153, 174, 189]

We generally assume that these defense measures will be implemented to state-of-the-art level and kept up to date as time progresses. Furthermore, we recognize that there are many similar lists available, however, we contribute additional benefit by categorizing them and discussing their efficacy during different stages of a cyber-attack. We do this by means of our generic model introduced in Section 4.4. Figure 64 illustrates this categorization and links our defense measures to the stages of the attack during which they apply.

The fields of efficacy are categorized as follows:

- *Account Defense:* The account defense specific defense measures deal with user management and password policies. They are illustrated in Figure 64 as a box, highlighted in purple, around the entire figure because they apply to all stages.
- *Social Engineering:* Social engineering specific counter measures, highlighted in yellow, comprise measures involving unsuspecting personnel and malicious insiders.
- *Host Integrity:* The host integrity specific measures, highlighted in blue,

deal with the security of physical devices e.g., servers, PC's, mobile phones, tablets, printers, and industrial computers.

- *Network Integrity:* Network integrity specific measures considers network security of the virtual environment. These measures apply to the section highlighted in red in Figure 64.
- *Network Detection:* These measures comprise network integrity and detection mechanisms that concentrate on reactive measures. They are also included in the red section.
- *Damage Containment:* Damage containment measures are useful in slowing, mitigating or stopping ongoing attacks for ensuring business continuity. They are illustrated in Figure 64 as a box around the entire figure, highlighted in gray, because they apply to all stages.
- *Recovery:* These measures are applied after an attack has concluded, for rebuilding affected systems and returning to normal operation. These measures are illustrated in green.

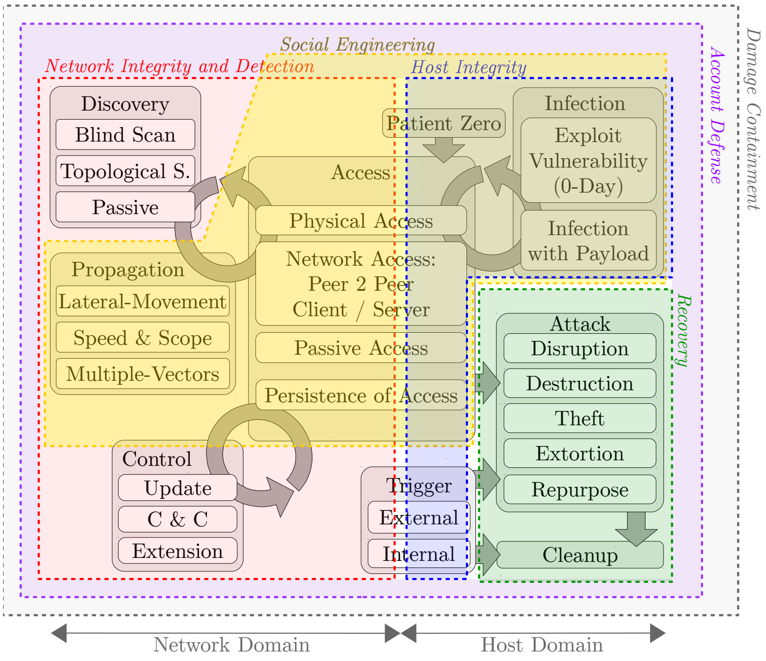


Figure 64: Efficacy of defensive measures during different stages of the cyber-attack life-cycle

Furthermore, we differentiate between proactive (preventative) and reactive measures. The latter including measures that take autonomous action (without human interaction) during an attack or attack-preparation. They are marked with superscript "R" in Table 37. All unmarked measures are proactive measures. In [182] Verizon shows that proactive security measures play a crucial role in keeping attackers outside the network premise as the attack and data exfiltration typically occurs within a very short time frame. However, the defenders generally discover that they were attacked only after the attack concluded. Therefore, we assume that the use of both proactive and reactive measures is of equal importance.

The second category includes the threat level, which implies the general attack methodology introduced in Section 4.5. Furthermore, Table 37 complements those measures already introduced in the security baseline, cf. Section 4.3. We cover and extend the security baseline in this list within the category "Basic".

- *Basic* requirements for standard security and basic operations should be implemented regardless of any suspected attack, cf. Section 4.3.
- *Pandemic specific* measures, cf. Section 4.5.1, should suffice to prevent pandemic-malware-level attacks.
- *Endemic specific* measures, cf. Section 4.5.2, should suffice to prevent endemic-malware-level attacks.
- *Contagion specific* measures, cf. Section 4.5.3, should suffice to prevent contagion-malware-level attacks.

Table 37: Defensive Measures for Smart Grids, published in [43]

Field of Efficacy	**Threat Level**	**Measures** based on: [2,8,11,20,26,28,44,49,59,66,69,71,80,81,84,104,113,114,122,124,125,127,144,152,153,174,189] Notation:^{R}= Reactive security measure
Account Defense	Basic User Management	2-factor authentication when a user is about to perform administrator work or wants to access sensitive information in critical networks, e.g., smart card + password
Account Defense	Basic User Management	2-person confirmation on critical processes, e.g., four eye principal on switching processes
Account Defense	Basic User Management	Administrator access: Disable local administrator accounts for the user, preventing credential theft and propagation
Account Defense	Basic User Management	Administrator access: Ensure strong 2-factor authentication and enforce the use of strong passwords
Account Defense	Basic User Management	Administrator access: Limit administrator accounts to a small number of persons and monitor these accounts, cf. incident cartoon on https://xkcd.com/838/
Account Defense	Basic User Management	Administrator access: Limit remote connections only to personnel that really need them and monitor them. Use 2-factor authentication on remote connections.
Account Defense	Basic User Management	Administrator access: Regularly audit your accounts on “who requires what rights?”
Account Defense	Basic User Management	User management: All users, including administrators, are restricted to access and environments suitable for their role.
Account Defense	Basic Hardening	Force manufacturers to disclose company backdoors and maintenance access upon signing the contract. They need to be either closed, or in the worst case, heavily protected
Account Defense	Basic Education	General employee training on awareness, the dangers of the Internet, and how to defend against them
Account Defense^{R}	Basic Hardening	Login failure: Implement increasing password window size after failed login attempt to counteract timing- and brute force attacks
Account Defense	Basic Hardening	Login failure: Monitor login attempts and corresponding inbound connections
Account Defense	Basic Password Policy	Change default passwords immediately, or lock devices that cannot be modified into separate VLAN’s
Account Defense	Basic Password Policy	Do not allow password reuse; Also forbid iterations of very similar passwords, e.g., Georg_12 and Georg_13
Account Defense	Basic Password Policy	Do not allow shared passwords, such as shared password safes, lists, or shared accounts
Account Defense	Basic Password Policy	Do not force people to change the password every 3 months, it leads to weak passwords or personnel write it on paper that is glued to the monitor
Account Defense	Basic Education	Employee training on not to use the same private and work related passwords (training time may be perceived as annoying)
Account Defense	Basic Password Policy	Enforce a password reset policy in the event of an attack, especially for VPN and administrative accounts
Account Defense	Basic Password Policy	Enforce strong passphrases covering complexity, length and expiry date, e.g., forbid single dictionary words
Account Defense	Basic Anti Virus	Prevent remote service creation that install or start a new local service
Account Defense	Basic Password Policy	Do not store Passwords directly as hashes but instead use PBKDF2, bcrypt, or scrypt which are made for storing passwords by iterating many hash operations
Social Engineering	Basic Education	Allow employee education on being able to say "NO" to access requests and report suspicious behavior
Social Engineering	Basic Education	Analyze social engineering attacks that have occurred against your organization

... Continued on next page

Table 37 Continued: Defensive Measures for Smart Grids, published in [43]

Field of Efficacy	**Threat Level**	**Measures** Notation:^{R}= Reactive security measure
Social Engineering	Basic Education	Business email compromise, i.e., CEO-fraud: Implement 2-person confirmation on transactions above a certain amount
Social Engineering	Basic Education	Create awareness with all employees for vulnerable network areas that are most critical
Social Engineering	Basic Business Continuity	Determine what resources attackers are most likely to pursue and those most critical to the business
Social Engineering	Basic Business Continuity	Determine where technology or processes can be altered to reduce or eliminate the threats
Social Engineering	Basic User Management	Determine where technology, policies, or company culture creates "soft spots" that are especially vulnerable to social engineering attacks
Social Engineering	Basic Education	Employee education on Internet threats, phishing emails, social engineering, weak passphrases, passphrase reuse, and unapproved USB devices
Social Engineering^{R}	Basic Business Continuity	Identify and isolate resources that are compromised
Social Engineering	Basic User Management	Implement behavior policies that make it easy for employees to perform secure actions without feeling rude
Social Engineering	Basic User Management	Perform background checks on employees and third parties
Social Engineering	Basic Business Continuity	Plan how to shut down an ongoing social engineering attack with the least amount of disruption to the business
Host Integrity^{R}	Basic Business Continuity	Consider active defense models for security operations such as the active cyber defense cycle [153]
Host Integrity	Basic Hardening	Disable auto mount on hosts
Host Integrity	Basic Hardening	Disable auto start on hosts
Host Integrity	Basic Hardening	Disable self-extract features on hosts
Host Integrity	Basic Hardening	Disable unused host system services e.g., JAVA, MS Office macros and unneeded extensions in browsers of viewer software
Host Integrity	Basic Hardening	Disable unused ports on clients and servers, e.g., USB or serial
Host Integrity	Basic Hardening	Disable unused services on server configurations with standard operating environment
Host Integrity^{R}	Basic Anti Virus	Endpoint security software from different vendors for servers because it primarily relies on up to date signatures to identify malware
Host Integrity^{R}	Basic Anti Virus	Endpoint security software using heuristics and automated Internet based reputation ratings to check a program's prevalence and its digital signature's trustworthiness prior to execution
Host Integrity	Basic Business Continuity	Ensure logging is enabled on host devices, servers, and operational assets, e.g., firewalls, switches, and routers
Host Integrity	Basic Hardening	Harden server configuration e.g., databases, web applications, customer relationship management, finance, human resources, and other datastorage systems
Host Integrity^{R}	Basic Firewall Setting	Host-based e-mail content filtering
Host Integrity	Basic Firewall Setting	Host-based web filtering: Restrict access only to permitted web sites who are work-related, e.g., block social media
Host Integrity	Basic Business Continuity	Implement reboot sequence for critical hosts in case the network is infected with memory resident malware. Run operations on fallback systems during reboot and cleanup.

... Continued on next page

Table 37 Continued: Defensive Measures for Smart Grids, published in [43]

Field of Efficacy	**Threat Level**	**Measures** Notation:^{R}= Reactive security measure
Host Integrity	Basic Business Continuity	Implement memory exploit mitigation, e.g., Enhanced Mitigation Experience Toolkit (EMET), that includes Data Execution Prevention (DEP) and Address Space Layout Randomization (ASLR), structured exception handler overwrite protection, and export address table access filtering
Host Integrity	Basic CIA	Implement hard drive encryption for critical hosts.
Host Integrity	Basic Business Continuity	Physical security for critical nodes and reporting of untrusted devices
Host Integrity	Basic Business Continuity	Physical security for field nodes and reporting of untrusted devices
Host Integrity	Basic Business Continuity	Physical Security: Make it hard for attackers to move around the premises, e.g., visitors are escorted at all times and cannot move freely, until leaving.
Host Integrity	Basic Business Continuity	Prioritize all your assets by business risk and criticality, as: "to protect everything is to protect nothing". Start to deeply implement security mechanisms on the most critical systems first, instead of “retrofitting it later”.
Host Integrity	Basic Network Segmentation	Prevent lateral movement by limiting workstation-to-workstation communication and Pass-the-Hash (PtH) attacks, e.g., implement private VLAN’s
Host Integrity	Basic CIA	Provide secure disposal of hardware and components, that no classified information can leak, e.g., shredding data
Host Integrity	Basic User Management	Remote field-switching devices: Allow only secure and up-to-date company owned devices in critical VLAN’s. Users cannot install APP’s with extended switching functions on private devices
Host Integrity	Basic User Management	Remote field-switching: Prevent device intrusion by 2-factors, e.g., fingerprint and password, and secure APPs with authentication for switching capability
Host Integrity	Basic User Management	Removable devices restrictions: Data-storage, data-handling, whitelisting allowed USB drives, encryption and shredding data
Host Integrity	Basic User Management	Restrict access to SMB and NetBIOS services running on workstations and servers where possible
Host Integrity	Basic Network Segmentation	Sandboxed trusted operating environment hosted outside of the organization’s internal network, for risky activities such as web browsing, e.g, virtual machines, chroot jail, segmentation
Host Integrity	Basic User Management	Smart phones: Apps for human resource management, e.g., holiday leave, should be separate from technical management and must NOT contain critical functions, e.g., remote switching. Use separate APP’s and devices for control functions
Host Integrity	Basic Network Segmentation	Smart phones: Restrict insecure private devices to office VLAN’s, seperate from critical VLAN’s
Host Integrity	Basic Firewall Setting	Software based application firewall, blocking incoming network traffic by default that is unauthorized or malicious
Host Integrity	Basic Firewall Setting	Software-based application firewall, blocking outgoing network traffic that is not generated by a whitelisted application, denying and logging network traffic by default
Host Integrity	Basic Update Policy	Updates (Patch management): Automatic application updates to prevent vulnerabilities
Host Integrity	Basic Update Policy	Updates (Patch management): Automatic operating system updates. Do not host old unsupported systems

... Continued on next page

Table 37 Continued: Defensive Measures for Smart Grids, published in [43]

Field of Efficacy	Threat Level	Measures Notation:^{R}= Reactive security measure
Host Integrity	Basic Update Policy	Updates (Patch management): Prioritize and patch known vulnerabilities based on the most critical assets in the organization
Host Integrity	Basic Update Policy	Updates (Patch management): Use hardware and software which encrypt and sign their security updates. Unsecure updates open vulnerabilities to hosts via Man-in-the-Middle (MitM) attacks. Do not buy cheap unsupported hardware for critical functions
Host Integrity	Basic Update Policy	Updates: Dimensioning hardware for future software updates: Smart grid devices remain in service for more than 10 years. Whenever resource constrained hard- or software is integrated into modern equipment as part of a modular design, the entire system security may be compromised with regard to sophisticated attacks. Therefore, these devices are prepared for future demands and provide sufficient resources to support updates
Host Integrity^{R}	Basic Firewall Setting	Use tools such as Fail2Ban or similar local services to automatically update firewall rules, that protect against Script Kiddies and low level brute force attacks
Host Integrity	Basic Firewall Setting	Whitelisting of very few Applications, i.e., permitted trusted programs, to prevent execution of malicious programs, to limit initial infection vectors
Host Integrity	Basic Anti Virus	Implement modern end-point-security software, that use heuristic and reputation based security services
Host Integrity	Basic Anti Virus	Workstation inspection for potentially malicious abnormalities e.g., using MS Office file validation or protected view feature. Alternatively, use Linux desktops and critical apps in chroot
Host Integrity^{R}	Pandemic	Email content passivation, e.g., redirect emails over a server that generates sanitized PDFs from attachments. Only deliver the original upon request and after malware check
Host Integrity^{R}	Pandemic	Honeypot and Decoying: Run dynamic analysis of content, including network traffic, new or modified files or other configuration changes inside a sandbox
Host Integrity	Pandemic	Principle of the least privilege: Grant access only to information and resources that are necessary for one legitimate purpose. Restrict and log everything else
Host Integrity^{R}	Endemic	Industrial computer Integrity: Confirm LOGIC integrity locally and also during updates (cf. PLC Blaster Worm)
Host Integrity^{R}	Endemic	Industrial computer Integrity: Confirm MEMORY integrity to ensure IDS, Code hooking detection, and data hooking detection
Host Integrity^{R}	Endemic	Industrial computer Integrity: Confirm the FIRMWARE integrity, locally and also during updates
Host Integrity	Endemic	Industrial computers: Never trust inputs, instead rely on additional independent channels. Attackers may fake control center data. Proof of concept in 33C3 Talk: Do as i say not as i do
Host Integrity	Endemic	Decentralization of critical services: Decentralization strengthens resilience against attacks. However, one must protect those decentralized critical services. Explicit countermeasures are required to counteract propagation methods not depending on a functioning network, e.g., infected removable drives
Host Integrity^{R}	Contagion	Intrusion detection and prevention on, e.g., clients, servers, and switches, identify anomalies during program execution, e.g., process injection, keystroke logging, and driver loading → IDS logs must be analysed, otherwise it is just an expensive heater

... Continued on next page

Table 37 Continued: Defensive Measures for Smart Grids, published in [43]

Field of Efficacy	Threat Level	Measures Notation:$^{\mathrm{R}}$= Reactive security measure
Host Integrity	Contagion	Smart phones: Buy the source code for APP's made for the utility company. Ensure there are no developer-backdoors
Network Integrity$^{\mathrm{R}}$	Basic Firewall Setting	Block spoofed emails using sender ID or Sender Policy Framework (SPF) to check incoming emails, and a "hard fail" record to help prevent spoofing of your organization's domain
Network Integrity	Basic Hardening	Configure servers to not answer ICMP services such as, e.g., traceroute, or ping. However, IPv6 requires some ICMP functionality
Network Integrity	Basic Firewall Setting	Create and maintain an inventory of relevant components that are in use, monitor their use and track unapproved components
Network Integrity	Basic CIA	Cryptographic Methods: Hashes and MACs for authentication: Use min. SHA-2 variants, e.g., SHA-256, SHA-384, SHA-51; Future implementations as fallback methods SHA-3, Blake2; Do not use MD5 and SHA-1
Network Integrity	Basic CIA	Cryptographic Methods: Symmetric Encryption: Use min. the block ciphers AES256-GCM or Camellia-GCM, and the stream cipher ChaCha20-Poly1305; Do not use RC4, A5/1 and A5/2
Network Integrity	Basic CIA	Cryptographic Methods: Asymmetric Encryption: Use min. DH 2048, RSA 2048, or better higher bitrates, for IoT use ECC; Future implementations and fallback methods mid term. 3072 bit RSA and long term 4096 bit RSA, and ECC-Curve25519; Do not use 1024 bit keys = equivalent of 80 bit symmetric encryption
Network Integrity	Basic CIA	Data lifecycle management: Evaluate if this data is still required after some time. Can it be deleted? Or does it increase the attack surface because criminals may want to get it? Consider, e.g., customer data which may be deleted after sending the bill. Publish deletion policy
Network Integrity	Basic Firewall Setting	Deny direct Internet access from workstations by using an IPv6-capable firewall to force traffic through a split Object Naming Service (ONS) server, an email server, or an authenticated web proxy server
Network Integrity	Basic Network Segmentation	Do not host critical devices in Internet facing networks, i.e., Demilitarized Zones or Perimeter Networks (DMZ). Build a dedicated infrastructure from the control center to the transformers because public Internet is easy to DoS. Use public Internet only for non critical services such as sending smart meter data. Do not use it for power switching equipment or smart meter enabled switching
Network Integrity	Basic Firewall Setting	Drop broadcast packets on clients and servers in critical networks, cf. in may 2013 Bavarian gas-grid control broadcast-messages pivoted to power grid SCADA and led to server failures
Network Integrity	Basic CIA	E2E encryption between field devices, substations, and control center, even on dedicated infrastructure to guarantee network and physical security
Network Integrity	Basic Business Continuity	Establish and maintain contact with authorities and security interest groups to be aware of vulnerabilities and threats
Network Integrity	Basic Business Continuity	Implement regular and continuous tests (table top exercises) to improve your defenses. Improve the incident response strategy accordingly. Scenarios should include gathering appropriate forensics, substation outages, manual operations, and recovering the SCADA environment.

... Continued on next page

Table 37 Continued: Defensive Measures for Smart Grids, published in [43]

Field of Efficacy	**Threat Level**	**Measures** Notation:$^{\mathrm{R}}$= Reactive security measure
Network Integrity$^{\mathrm{R}}$	Basic Firewall Setting	MitM attacks on IEEE C37.118 (NOT recommended [104]) require compromised local system requests and knowledge of configurations from PMU's to understand the data messages. Request packets are spoofed with the genuine recipient IP address. Such activity should be marked suspicious as a PMU has already provided configurations to the recipient.
Network Integrity$^{\mathrm{R}}$	Basic Firewall Setting	MitM attacks on IEC 61850-90-5 (recommended [104]) require an attacker to attempt disconnect a PMU from the key distribution center and then attempt MitM attack on communication between the PMU and Key Distribution Center (KDC). The PMU should raise an alert and detect gratuitous ARP packets, i.e., used for traffic diversion and raise an alert
Network Integrity	Basic Firewall Setting	Never place control outside the firewall, as is common practice by centralized IT departments, e.g., central IT department takes control over firewalls that protect critical sub networks
Network Integrity	Basic Firewall Setting	Properly tuned firewalls between network segments will give visibility into the environment and allow defenders the time required to identify intrusions
Network Integrity	Basic CIA	Protect sensitive information and critical services, e.g., user authentication, with services such as Microsoft Active Directory
Network Integrity	Basic Business Continuity	Regularly audit attempts to access sensitive company information. Include failed and successful attempts. (Who is allowed to do what?)
Network Integrity	Basic CIA	Remote field-switching: Communication to control systems through external networks, e.g., mobile carrier, must be protected through at least 2 separate layers of e2e encryption. One of which must be in the application layer, cf. OSI-model. Do not rely on standard 3G encryption, cf. [27]
Network Integrity	Basic Network Segmentation	Remote field-switching devices: Critical VLAN's must be restricted to secure APP's and company controlled devices. Otherwise, malware can enter through private networks
Network Integrity	Basic Update Policy	Secure remote updates in the DMZ with seperate credentials because extractor tools can read them from memory
Network Integrity	Basic Firewall Setting	Strict Firewall Rules with whitelisting, MAC filter, and dynamic traffic speed reduction unpon suspicion
Network Integrity	Basic CIA	Transport Layer Security (TLS) encryption between email servers to help prevent emails being intercepted and used for social engineering. Perform content scanning after email traffic is decrypted
Network Integrity$^{\mathrm{R}}$	Basic Hardening	Train defenders on using tools such as YARA to scan digital images and scan evidence collected from the environment
Network Integrity	Basic Business Continuity	Use a security development lifecycle for quality assurance
Network Integrity	Basic Business Continuity	Use external security audits to discover vulnerabilities, e.g., penetration-testing
Network Integrity	Basic Firewall Setting	Use IPv6 with a suitable address concept (RFC4864) instead of network address translation. NAT is used in IPv4 networks to hide devices behind a gateway and is not recommended in IPv6 as it obstructs applications, e2e transparency and network layer security association
Network Integrity	Basic Network Segmentation	Utilize VPN's for segmentation

... Continued on next page

Table 37 Continued: Defensive Measures for Smart Grids, published in [43]

Field of Efficacy	**Threat Level**	**Measures** Notation:$^{\mathrm{R}}$= Reactive security measure
Network Integrity	Pandemic	Ensure that devices in the network architecture, e.g., switches, are managed and have the ability to capture data from the environment to support passive and active defense mechanisms
Network Integrity$^{\mathrm{R}}$	Pandemic	Intrusion detection and prevention: Signature and heuristics based identification of anomalous ingress and egress traffic crossing network perimeter boundaries
Network Integrity	Pandemic	Secure architecture: Segmentation of networks and clear, strong restrictions between office and control network
Network Integrity	Pandemic	Utilize the latest encryption and not the minimum some standards may permit. Standards often allow methods that are easily crackable
Network Integrity	Endemic	Protect/harden your Active Directory service as it has vast privileges on authentication inside the domain. It is a blinking target for APTs. See Enhanced Security Admin Environment (ESAE) → Tier 0: Enterprise Admins, Tier 1: Server/Cloud and application Admins → Tier 3: Administrative control of workstations and devices
Network Integrity	Endemic	Port based network access control, e.g., implement a policy on how to initially access the network
Network Integrity	Endemic	Use air gaps. However, be aware, removable drives can form bridges, cf. Stuxnet
Network Integrity	Endemic	Use jump servers for intermediary access of external agents, to devices behind a firewall. The Jump Server checks credentials, uses time-based one-time passwords and provides the information needed to communicate with the target devices
Network Integrity	Contagion	Bug-bounty programs could help discover vulnerabilities
Network Integrity$^{\mathrm{R}}$	Contagion	Anomaly based IDS on ingress and egress traffic across the network perimeter, e.g., packet size, unsolicited traffic, and connection attempts, e.g., RST, ICMP, SYN, and DSC
Network Integrity	Contagion	Mistrust third parties as they may, unknowingly, act as entry vectors. According to European NIS rules, large actors must disclose attacks to help mitigate them in future. Small actors are exempt from these rules, therefore posing a threat when involved in a secure network
Network Integrity$^{\mathrm{R}}$	Contagion	Plausibility checks of IO data, including independent sensing, to run comparison on expected process state information, against physical destruction, e.g., Stuxnet manipulates outputs
Network Detection$^{\mathrm{R}}$	Basic Firewall Setting	Email content filtering, allowing only whitelisted business related attachments. Preferably analyze, convert, and sanitize hyper-links and attachments
Network Detection$^{\mathrm{R}}$	Basic Business Continuity	Ensure analysis personnel has access to technologies such as isolated test quipment, e.g., sandboxes, to quickly analyze phishing emails or odd files and extract indicators of compromise to search infected systems
Network Detection	Basic Network Segmentation	Reduce the number of active services to a minimum and use only what is really needed, e.g., office networks require email telephony and print services, critical ICS networks do not
Network Detection$^{\mathrm{R}}$	Basic Firewall Setting	Utilize web DNS reputation via commercial feeds that rate the trustworthiness of domains. Redirect dangerous web requests to a warning page
Network Detection$^{\mathrm{R}}$	Pandemic	Dependency graphs for damage assessment
Network Detection	Pandemic	Centralized time-synchronized logging of allowed and blocked network activity, with automated immediate log analysis, storing logs for at least 18 months and data aggregation

... Continued on next page

Table 37 Continued: Defensive Measures for Smart Grids, published in [43]

Field of Efficacy	**Threat Level**	**Measures** Notation:^{R}= Reactive security measure
Network Detection	Pandemic	Centralized time-synchronized logging of successful and failed events. Including automated immediate analysis and data aggregation
Network Detection	Pandemic	Configure your IDS such that rules can be deployed quickly to search for intruders
Network Detection^{R}	Pandemic	Implement alarm package priorities for abnormal events within the control network
Network Detection^{R}	Pandemic	Information flow analysis: Attack graphs for vulnerability analysis, e.g., subset of flow analysis
Network Detection^{R}	Pandemic	Information flow analysis: Full traffic enables analysts to view system behavior, e.g., NetFlow to discover scanning
Network Detection^{R}	Pandemic	Perimeter control on unsolicited outgoing traffic on the border gateway of critical networks
Network Detection	Pandemic	Train defenders to hunt for odd communications leaving the critical environment, e.g., new connections, unsolicited connections, new domain requests, and suspicious packet size
Network Detection^{R}	Pandemic	Use honeypots and a darkspace to discover unsolicited traffic
Network Detection	Pandemic	Web domain white-listing is more thorough than blacklisting
Network Detection^{R}	Endemic	Attack trend analysis in the public Internet for early detection of intrusion vectors
Network Detection^{R}	Endemic	Detect the top 10 DNS-resolutions from yesterday that have NEVER occurred in your critical network. Decide to block or quarantine them
Network Detection^{R}	Endemic	IDS between trusted networks: Gateway protection over the top level network is insufficient. Once an attacker is inside, propagation cannot be counteracted, i.e., Sony hack Nov 2014. Access vectors may be trusted partner networks such as external services, e.g., data-centers, security services, and parent- or subsidiary-companies
Network Detection^{R}	Endemic	Passive measures alone are no longer an appropriate stand-alone-defense against endemic malware, e.g., single implementations of air gapped networks, firewalls, anti-malware tools on the ICS, and topology changes. Consider that automated detection and a combination of defenses including human defenders are necessary.
Network Detection^{R}	Endemic	Plan and implement an intrusion response.
Network Detection^{R}	Endemic	Use event monitoring, configured and monitored specifically for high-value ICS SCADA systems.
Network Detection^{R}	Contagion	Causality analysis and forensics.
Network Detection^{R}	Contagion	If an APT at work is detected, do NOT block its traffic. APTs have redundancies, alternative RATs, and heartbeat services. First, slow specific traffic, e.g., replay attacks; then, secure your system and setup surveillance. Once you feel confident, or critical devices are about to be compromised, block the attacker. Estimate at least one week for mitigation.
Network Detection^{R}	Contagion	Intrusion and anomaly detection in every subnet record legitimate traffic patterns to learn what could be suspicious behavior, e.g., hidden channels.
Damage Contain.	Basic Business Continuity	Build up competences, i.e., emergency response teams, inside the firm.
Damage Contain.	Basic Decentralization	Create fallback systems which rely on little or no ICT. Include them in cyber-attack scenarios.
Damage Contain.	Basic Decentralization	Guarantee emergency bandwidth for industrial control networks in shared infrastructures, e.g., dropping non-critical connections for control network has priority.
Damage Contain.	Basic Decentralization	Over-provisioning of resources, e.g, redundancies for failure scenarios.

... Continued on next page

Table 37 Continued: Defensive Measures for Smart Grids, published in [43]

Field of Efficacy	**Threat Level**	**Measures** Notation:$^{\mathrm{R}}$= Reactive security measure
Damage Contain.	Basic Firewall Setting	Smart meter central management: Allow max. 1 remote disconnect event per minute in the entire smart meter network, i.e., prohibit mass switching and cascading effects.
Damage Contain.	Basic Hard- Software	Smart meter firmware: Prohibit rapid switching on field devices that are capable of managing heating, cooling, or air-conditioning. Only allow 1 switching event at a random time during the next 15 Minutes to mitigate cascading effects.
Damage Contain.	Basic Education	Train employees in cyber-attack response plans, incorporate IT and operations personnel.
Damage Contain.$^{\mathrm{R}}$	Pandemic	Honeypot and decoying: Allow infection of fake hosts, and simulate a network of targets, e.g., HoneyD can emulate large networks. However, anti honeypot methods exist.
Damage Contain.$^{\mathrm{R}}$	Pandemic	Slow down suspicious traffic at first parameter threshold violation to avoid false positives, then isolate traffic source and block at second threshold.
Damage Contain.	Pandemic	Use separate hardware interfaces for office and critical networks because Denial-of-Service (DoS) attacks on one network may compromise the shared hardware of others.
Recovery	Basic Business Continuity	Backup 1: Set a secure baseline configuration with standard images that provide fallback capability including secured applications and operating system configurations, e.g., Debian (stable) 8.7 Jessie + approved tool-chain.
Recovery	Basic Business Continuity	Backup 2: Short term, permanently mounted, incremental data backups against low level cryptolockers.
Recovery	Basic Business Continuity	Create disaster recovery plans from cyber-attacks, that include redundancies in critical equipment. The Ukraine attack 2015, cf. [56], shows that recovery processes can take months, if unprepared.
Recovery	Endemic	Backup 3: Long term redundant data backups, i.e., not permanently mounted, against high level cryptolockers. Keep backups offline when not needed. Consider that attackers may implement encryption to Network-attached Storage (NAS) and only after a period remove the access keys.
Recovery	Contagion	Backup 4: Backup critical software installers and control logic, e.g., ICS applications, and include hash-sums to harden them against tampering.
Recovery	Contagion	Backup 5: Take digital images of critical assets, e.g., IED configurations and SCADA firmware, every 12 months and record a change-log when installing upgrades.

Concluded

11 Conclusion

Notice of adoption from previous publications:
Parts of the contents of this chapter have been previously published in the following papers:

[44] discusses the conclusions published about our topological analysis, including the results, benefits, and drawbacks of different network topologies concerning their application in smart grids. These results were also presented in [37, 39].

[48] discusses several attack types specific to smart grids. We conclude them in this section and discuss future developments, which were also presented in [38, 40, 45].

[43] discusses defensive measures that build on the available attack types, resulting in a list that utilities may use to improve their defenses. This list was also presented in [41] and is shown in Section 10.

[47] discusses the implications monocultures can present in critical infrastructure networks.

[46] introduces several metrics that define the attack efficiency and defendability upon which we evaluate our malware simulations. We conclude the results here.

In this chapter we conclude our results, discuss an outlook, and future improvements.

11.1 Smart Grid ICT Topologies

This work provides the theoretical basics on ICT topologies for critical infrastructure networks, including a theoretical evaluation for urban smart grid environments, cf. Section 7. We present *benefits and drawbacks of four ICT topologies*, i.e., one fully centralized and three hybrid topologies based on quality indicators including resource control, security, resilience, quality of service, compatibility, and cost. Although we find that centralized and decentralized topologies have benefits in some respects, the mesh and cell topologies overcome most of their shortcomings.

The *centralized topology*, although effective for retrofitting current power grids to smart grids, shows that malware infections can immediately reach the central control node (HL-node), when defensive measures fail, which requires the *extensive protection* of the HL-node, including *several redundancies with fallback strategies*, to fend off malware attacks. Otherwise, these resources could be distributed over field nodes, implementing early warning systems when they are attacked, which is discussed in the following topologies.

The *cell topology* provides the most benefits for future smart grids through the *placement of sensitive nodes in physically secure and controllable locations* and through the hierarchical structure from the field level to the cell controllers (ML-nodes). This placement allows for *neuralgic nodes* that in turn are required to operate most the defensive measures. To implement this topology, electrical open rings would have to be upgraded to microgrids with corresponding communication nodes. Furthermore, it requires upgrading the transformer rooms which house the ML-nodes with surveillance equipment to physically secure the locations. This allows for situational awareness and control over the substations and their respective subjacent LV-grids. The mesh network between the local cell controllers yields benefits for congestion management and network resilience. However, security measures on each node and physical security for the ML-nodes must account for the containment of malware. Therefore, *intelligent mitigation measures in the field will be required.*

The *mesh topology* provides extensive propagation paths in the LL-ICT and ML-ICT level. Although these meshes allow for *increased resilience*, the attacker can also use those *alternative paths and optimize the attack strategy.*

The same is true for the *decentralized topology*, with the addition that the HL-node is also included in the mesh network, making *host-based defenses even more important.*

Since utility companies may prefer to purchase key-ready systems in order to decrease costs, the reuse of hardware and software in these systems may replicate security vulnerabilities across hierarchy levels, thus, create a monoculture, as discussed in [47]. Network segmentation within the ICT topologies and the progressive implementation of firewalls, strict rules, and anomaly detection, can prevent the propagation of malware in such an environment. Additionally, penetration testing can help to discover vulnerabilities in the host devices.

In accordance with research question 1, we investigated which communication topology delivers the most promising features in terms of security by design. Since communication networks can never be 100% secure, we investigated several network topologies with the goal of discovering features that can complicate cyber-attacks. However, our theoretical works show that none of the topologies can reach an acceptable level of security by default, without additional defensive measures, cf. Section 10. We have shown that the cell topology has the best features in containing malware because it does have some of the benefits of the resilience of mesh networks, however, also the benefits in terms of propagation containment because malware attacks cannot move freely to neighboring nodes in this restricted environment, thus, segmentation. The cell topology requires ML-nodes to be upgraded with the latest defense measures and local intelligence providing some decentralization. This additional layer of ML-nodes acts as a firewall toward the HL-node which is only possible due to the increased intelligence in these ML-nodes.

11.2 Malware Comparison

In accordance with research question 2, we investigate several malware models, ranging from aggressive to stealthy types, cf. Section 8.1. In total we investigated 19 types of malware with respect to metrics introduced in our *generic cyber-attack model*, i.e., propagation mechanisms, detectability, targets, persistence, and countermeasures. We analyzed their relevance to smart grid attacks and provided *3 hypothetical superclasses of malware* that represent malware types which in future could target smart grids, namely, pandemic, endemic, and contagion malware. Our aim is to *raise awareness on the defenders' side* of the need to build defenses against, at the very least, the endemic malware class. Since attackers only need to exploit one particular vulnerability, while defenders have to defend against all potential vulnerabilities, an asymmetry in resources and knowledge is apparent.

We find that many modules covering different areas and attack vectors discussed in our generic attack model, cf. Section 4.4, can be acquired on the black market today [134]. Yet, assembling them into a functioning attack platform is something else entirely, requiring skill and knowledge on the inner workings of the network structures in utility companies. However, recent attacks [56] on utility providers show that these resources, despite their high level, are in fact being applied. The more knowledge on malware and different modules is available, the lower the implementation threshold becomes, giving less equipped attackers highly capable tools that require high defensive effort on the utilities part.

As discussed in Section 8.2, the most *significant trends in future malware creation have been moving toward stealthiness at the cost of speed.* Additionally, as expected, *modern malware shifts away from simple, aggressive types towards complex, modular and sophisticated ones* with a considerable set of capabilities. Recently, such malware types have been heavily involved in data theft campaigns. However, there are also several precedents for cyber-physical attacks and an increasing trend toward highly versatile malware.

The complexity is ever increasing as demonstrated by the evolution of members of the same malware family within few years, for instance Stuxnet (6 modules) and Duqu2 ($>$ 100 modules). Duqu2, among others, has gained many functions including new exploits, data theft-, persistence-, and propagation methods. The addition of one single module could enable such real world examples to launch a cyber-physical attack. The required technology being readily available, we consider this trend to be of particular concern for utility companies, and encourage them to raise the bar for attackers.

Pandemic malware, cf. Section 4.5.1, with its simple attack methods may be defeated by standard IT-security measures found in best-practice guidelines, c.f. Section 4.3.

Endemic types, however, require more scrutiny on the defenders side. This can include anomaly detection and specific employee training, among others, cf. Section 4.5.2. Furthermore, we consider *endemic malware to be the current state of the art* and, therefore, the most immediate threat with many different modules/capabilities for sale on the black market, that can be implemented by less equipped attackers.

Contagion malware, cf. Section 4.5.3, is an even greater challenge due to its increased stealthiness and persistence. It may in part be counteracted with the same technologies required for the defense against endemic types. In addition to more scrutinizing network segmentation, anomaly detection, and honeypots could be feasible approaches. Contagion malware revolves around increased stealthiness beyond the capabilities of endemic malware. Therefore, we expect to see contagion class malware only from highly skilled and financed sources, thus, rarely. However, since many defense mechanisms are required anyway for the defense against endemic malware, the contagion malware should be considered an advanced threat that requires only some more attention beyond the currently required security standards that can defeat endemic malware.

Finally, zero-day vulnerabilities offer an attack surface that utility companies cannot be expected to counteract. However, with proper update- and segmentation policies, IDS at several checkpoints, and strong organizational processes, defenders may prevent social engineering and sabotage. Appending on those security measures, we urge utility providers to diversify their stock of devices and security implementations, i.e., to ensure heterogeneity in order to mitigate malware spreading that exploits one specific vulnerability, cf. [47].

Summarizing, the main findings of the theoretical malware analysis are:

- The *generic life-cycle model* formalizes the stages of malware-based cyber-attacks and enables us to investigate existing malware by dissecting it into recurring cycles (cf. Section 4.4). This allows a detailed comparison of characteristics in existing malware and provides a useful basis for predicting future developments.
- The *investigation of existing malware* shows a clear trend toward in-

creasingly stealthy cyber-attacks, with the goal of infiltrating highly secure networks, cf. Section 8.1. This trend should be a warning for utility companies to extend their defensive capabilities.

- The *prediction of potential future attacks*, cf. Section 8.3, based on the analysis of existing malware leads to three conceptual models, cf. Section 4.5, which can be used as basis to develop defensive strategies.
- The *mapping of security measures* to all future attack vectors assesses the usefulness in mitigating specific attacks by a qualitative assessment, cf. Tables 12 and 30. Furthermore, the mapping identifies where gaps still exist that can be exploited by future malware.

11.3 Metrics

We introduce new metrics used in our network simulations concerning the malware types, cf. research question 3, such as connection attempts, protocols, spreading patterns, or target finding, which have considerable influence on the infection rates. We *develop these effective metrics for the detection of malware and containment measures against cyber-attacks.* Furthermore, our novel generic model of a malware life-cycle analysis includes a comprehensive comparison of existing attack methods, several detection metrics, and an outlook on smart grid specific attacks.

11.4 Malware Propagation and Attack Resilience

We discussed our simulation results in the Sections 9.2, 9.3, and 10, which outline on the one hand; The repercussions arising from *weak security functions inside monocultures.* There, we show that vendors and utilities have huge incentives to implement their control infrastructures as monocultures in terms of both, hardware and software. The same hardware is mainly used to lower development-, deployment-, and replacement costs. Using equal software decreases operational and maintenance costs. However, the huge number of identical nodes in such networks supports fast and efficient propagation of malware once a vulnerability is found.

Our analysis confirms that monocultures support fast malware spreading, in particular if the communication networks are not configured and segmented properly. As this can have catastrophic consequences, critical networks must not be connected to shared network resources like enterprise networks unless absolutely necessary, with financial benefit not being a valid enough reason in our opinion. Advanced security measures, e.g., anomaly detection are recommended to be implemented on neuralgic nodes, where possible, to detect and prevent malware from infecting neighboring networks.

Furthermore, we discuss the general simulation results of our simulation model including the metrics leading to *calculating the attack-defendability* for each case and several defensive measures alongside to support our discussion. This represents our work on research questions 1, 2 and 4, whereas 1 represents the network topology, 2 represents the malware propagation, and 4 represents the attack-defendability.

Our simulation results on malware propagation include 12 simulation sets of three malware types over four topologies. The results in Section 9.3.5 summarize them and *support our discoveries on why the widespread application of stealthy malware is increasing.* We discovered *no significant drawback of slow and stealthy malware over fast and noisy malware.* It rather seems that the implementation of better security measures in the recent past have forced malware developers to increase development efforts to maintain their advantage by obfuscating among native traffic, but with covert capabilities on the rise.

The contagion malware proves that *hidden propagation provides additional benefit* at the cost of even more speed, and increased development effort. Although there are increased resources required for initial development, advanced types become public, or are sold in time, thus, less equipped attackers can modify them and implement their own interpretations. Many *additional benefits lie in the host-based attack capabilities of advanced malware*, e.g., new infection vectors, increased persistence, and attack versatility making advanced malware even more challenging to detect.

Concerning the *centralized topology*, we conclude that the *pandemic malware*, although being fast and noisy, has the benefit of fast infection of the control center, enabling it to quickly launch a destruction attack. However, we assume that utilities will implement basic security measures to state-of-the-art level, thus, pandemic malware may not be able to challenge the ICT security of modern smart grids. In case it can, it outruns human intervention within its < 1 s attack window. *Endemic malware* on the other hand, is much harder to detect but shows similar infection durations compared to the pandemic malware. However, its initial infection curve rises slowly, allowing some reaction time for defenders, and especially for automated defenses. We assume that this malware type has advanced on-board capabilities, thus, requires a well selected set of defensive measures to be defeated by. The *contagion malware*, on the other hand, is the most difficult to detect, thus, the attack efficiency is high even though it shows extremely slow infection. This fact makes the contagion malware the overall benefactor at the cost of high development costs.

The *cell topology*, which provides more options to bypass congested links, shows some benefits on the resilience of the nodes. However, these benefits also help malware propagate. *Pandemic malware* propagates quickly, however, produces much network noise, thus, it should be easy to defend against. Because it can move through a mesh network, the infection times increase slightly compared to the centralized topology. *Endemic malware* optimizes its propagation pattern and slightly surpasses the attack-efficiency of pandemic malware. *Contagion malware* can, due to the hidden propagation, achieve the highest attack-efficiency because it is near undetectable in network traffic. However, it has the slowest infection ratio, at the highest sophistication level of on-board features.

The *mesh topology* utilizes mesh networks on a large scale in the LL-ICT and ML-ICT level. All malware types, except contagion malware, can utilize these alternative routes. Contagion malware never uses alternative routes intentionally, only when packets are routed through the mesh network as part of legitimate traffic. *Pandemic malware*, has a higher attack efficiency compared to endemic malware which only slightly surpasses it. However, *endemic malware* can recover some drawback from the larger payload by increased infection efficiency and scanning efficiency. *Contagion malware* has a nearly perfect propagation efficiency which is due to its extensive obfuscation capabilities.

The *decentralized topology* utilizes mesh networks throughout all hierarchy levels. Therefore, it is the most difficult to implement in existing power grid control systems. Moreover, it would require complete retrofitting without implementing any of the existing structures. Its extensive use of mesh networking including many alternative links increases the resilience of the network. However, malware also may use this feature for optimized propagation. It is also the most difficult case for containing malware because there are always redundant paths available that cannot all be controlled. *Pandemic malware* shows the lowest attack efficiency which is mainly caused by the aggressive scanning and the low infection efficiency. *Pandemic malware* shows better attack efficiency compared to the endemic malware in this topology. However, the infection efficiency is in both cases very low. *Contagion malware* follows a hidden propagation strategy, resulting in slow infection. However, its medium attack efficiency provides benefit over both, pandemic and endemic malware. On a positive note, contagion malware must, in this case, travel over the backhaul infrastructure. This allows for better containment options compared to the other malware types. However, since contagion malware operates covertly, the detection systems that are likely to be implemented in a decentralized topology, that favors no ML-node over another, may struggle detecting this malware type.

11.5 Defensive Measures and Containment Capabilities

Section 10 *introduced a list of recommendations for defensive measures* against different malware types. We remark that this extensive list cannot be quantified in our simulation model, thus, we take a theoretical approach.

We conclude from our simulations that network segmentation should rank high among the most important defensive measures. Alongside that, we also suggest to protect the gateway and perimeter nodes that face other network segments with additional host-based measures, e.g., standard hardening guidelines to achieve better host security. We advise to *force all communications through check points* which require up-to-date IDS and strong firewall rules, e.g., whitelisting, for perimeter-IDS. Furthermore, protecting the gateways may not suffice when identical vulnerabilities are available to attackers via alternative propagation channels. Admittedly, monocultures are easier to manage in terms of administration, however, they also present attackers with a known environment. Our simulations consider the initial propagation phase of malware, thus, automated movement without C&C interference. Therefore, *monocultures can significantly support horizontal and vertical propagation* unless automated defenses, i.e., reactive measures, apply remedies.

Since preventative measures are very useful as a deterrent against attackers, they are often not suitable for the detection of ongoing cyber-attacks. They are, however, no less important in defending critical infrastructures. According to Sun Tzu [178], subduing an enemy without fighting is the highest form of war. Therefore, we suggest to *consider well prepared defenses that may discourage most attackers*, that may not be highly financed. However, defenders are at all times at a disadvantage against attackers by having to prepare against all possible attack vectors, whereas an attacker only requires one vulnerability. On the other hand, defenders can use a home field advantage to their own terms and lead attackers to approach on their own accord, thus, defenders can mislead attackers. Although prevention is good, awareness is better, thus, we recommend Section 10 as a general checklist of items worth implementing in the security concept of critical infrastructure communications. We suggest that detection mechanisms should be implemented early, since they can best increase the level of defense, e.g., IDS, honeypots, or network event monitoring. These are predominately found in the categories "Host Integrity", Network Integrity", and "Network Detection". Furthermore, several measures, e.g., honeypots or darkspaces, can be used to actively mislead attackers. All war is based on deception, thus, appear weak where you are strong and defend what is weak [178].

During the final stages of a cyber-attack, those measures in the category "Damage Containment" become relevant. They are, for the most part, meant to slow or stop an ongoing attack. After clearing all malicious code, recovery can commence until full functionality is restored. We do not claim completeness of the list in Section 10, as new measures can arise and be implemented at any time. The list should be extended by knowledge collected from new attack detections. We do, however, consider it the current state of the art (in the year 2018). These measures are categorized by efficacy in those stages that occur during a cyber-attack and the threat level different malware types represent. This list can be used by operators of critical infrastructures to improve their defenses. It can, if implemented, help utilities to discourage even powerful attackers. Furthermore, the measures are designed to consider all types of attackers and implement some measures that may increase the cost of attacking (the resources required) to a possibly unfeasible level even for APT's. Section 10 provides an explanation on how to use the list aside from extra information on the containment properties of those measures.

In accordance with research question 4, cf. Section 8.1, we investigated how to increase the attack-defendability and attack-resilience of critical communication networks. We introduced the metric *attack-defendability* as a measure for calculable security by design, cf. Section 6.13. Furthermore, the metric *attack-resilience* expands upon additional defense measures which cannot be quantified in our simulation environment. Therefore, we extend the concept of *resilience against random failure* by targeted cyber-attacks, namely *attack-resilience*, cf. Section 6.16. This metric satisfies our needs because cyber-attacks do not occur randomly, but represent targeted- and deliberate failures rather than random failures. Therefore, we cannot rely on mere redundancies, but instead rely on the proactive and reactive counter measures as discussed in Section 10. We satisfy this content with a collection of proactive and reactive measures to achieve a higher level of security for critical infrastructures.

11.6 Outlook

In this section we discuss an outlook beyond this work, i.e., future requirements to defend against our three malware classes. We include the following which could not be covered within the scope of this work: [44]

- *Adjustment:* Knowledge collected in future attack incidents has to be incorporated in predictions of upcoming malware. For instance if malware with a novel spreading or scanning method is observed, this information has to be shared and included in future defense strategies for future threats.
- *Information sharing:* Cooperation is an important defense strategy. Incidents should be reported. Detailed information about attacks and

attack preparation should be shared in the community. It may be required to provide incentives and technical solutions for controlled data sharing or even to enforce it to resolve concerns in organizations about sharing information with potential competitors.

- *Lightweight detection:* New lightweight detection methods are required to separate legitimate from illegitimate network traffic without exceeding resource demands and costs. They should guarantee long term functionality and maintenance in devices that operate over periods of more than 10 years.
- *Hiding techniques:* Malware may use sophisticated hiding techniques such as covert channels to prevent detection of malware communication. Future detection systems need to prepare for this and should incorporate covert channel detection methods.
- *Cyber-physical systems research:* Further research into a holistic view on the coherences between cyber-physical systems provide better understanding of the processes to minimize the attack surface and harden them against attacks. Here especially the interrelations between different systems is relevant for potential attack spreading.

11.7 Improvements

In this section we discuss several improvements that could be implemented in future works:

- We find that our model, although capable of simulating a large number of nodes, lacks performance, i.e., simulations with more than 500 nodes may take several days to complete, as illustrated in Section 9.1.2. This could be optimized by additional code reviews, however, optimization must utilize multi core processing.
- Concerning multi core processing, one caveat remains within the ns3 simulation environment. ns3 is only capable of multi-core processing on P2P-links, i.e., as soon as other link technologies are used, e.g., mesh networks, multi-core processing is not implemented at a basic level of the simulation environment. However, our scope is not to optimize ns3, which is a task left for the core developers, thus, we do not use multi core processing.
- The random factors implemented in our simulation were disabled to meet our goal of focusing on detection metrics, efficiency metrics, and countermeasures. However, large scale statistical analysis may be possible upon implementing the above mentioned performance optimization, increasing the value of the simulation model. For the results to

remain representative within this simulation set, we constrain our 12 simulations to the same parameters without the random factors.

- Our metrics do not consider C&C traffic because we consider the initial stages of malware attacks, thus, it is out of scope for this work. We investigate the initial phase of malware spreading, effective counter measures, and defensive strategies. However, C&C traffic can be implemented in future works that test C&C traffic inside our four topologies and three malware types. This would expand the scope of the model from the initial spreading phase to the operation phase, as illustrated in Figure 16. However, we limit our approach to the attack efficiency and attack defendability metrics, cf. Sections 6.12 and 6.13, that represent metrics we can attribute with calculable output from our simulation model.

- We developed several additional metrics that extend the scope of our model, namely, attack containment and attack resilience. However, since they must include C&C, we cannot quantify them in our simulations, cf. Section 6.14. We do discuss them in theory and provide formulas for their calculations, however, a full analysis of all defensive measures would require the attribution and consideration of all defensive measures, cf. Section 10, for each malware type in each strategic point and in all topologies.

- All simulations in Section 9.3 could be repeated for all other attack methods, e.g., data theft attacks or extortion schemes. These were introduced in Section 4.4.6 and include additional discussions and defensive solutions in our theoretical results, cf. Section 8.3. However, we limit our simulation model to the most sinister attack, namely the destruction attack, due to its significant impact on society. We do not simulate the other attack types because they do not represent such an immediate need of protection. Furthermore, the other attack types can be considered a subset of the destruction attack because an extortion and data theft campaign can be conducted prior to the destruction attack.

- Since the most significant limitations have occurred with the mesh network protocol (OLSR) and its maximum network size, cf. Section 9.1.2, we suggest to check the utilization of other protocols in future. In order to accomplish very large single mesh networks with hundreds of nodes, these limitations must be overcome with either hybrid infrastructures or new mesh protocols.

- A_{flow} represents covert anomalies inside legitimate network traffic. However, typical anomaly based IDS can only investigate network flows, and thus, packet content, as long as they are not encrypted.

  We understand that E2E encryption is nowadays widely implemented, especially for control communication in cyber physical systems. Therefore, the A_{flow} method can only be used in cases when the anomaly detection system is allowed to act as a man in the middle for encrypted traffic, which is an existing concept on the market of IDS-manufacturers. However, this is compromising E2E security. The author, however, regards man in the middle IDS systems critically.

- We note that patient zero could be set to any node position in our simulation model's hierarchy. However, we simulate all test-cases with LL-nodes as the starting point because then the simulation represents the longest propagation path, thus, we can observe all defined infection durations. If, for instance, am ML-node were chosen for patient zero, $T_{first.GW}$ would occur immediately, as introduced in Section 6.3. The same is true for direct infection of the central control node, which invokes $T_{C.Center}$ immediately, and so on.

Final note.

We conclude that, despite the heavy complexity of malware defense, utilities can reach a good level of security with just a few well implemented technical measures and optimized organizational processes. On top of these, additional measures can harden a control network further against more advanced threats, albeit, 100% security is never achievable. Moreover, security measures should not just be implemented and then left alone, but instead be audited, maintained, and managed continuously. Therefore, utilities need to establish a permanent security staff, equipped with enough resources to counteract future threats.

Bibliography

[1] 3GPP A Global Initiative. The Mobile Broadband Standard. [Online] Available: http://www.3gpp.org/technologies, April 2016.

[2] A. Abbasi and A. Majid. Do as I Say not as I Do: Stealth Modification of Programmable Logic Controllers I/O by Pin Control Attack, Chaos Chommunication Congress 2016, Hamburg, Germany, December 2016.

[3] F. Akhtar and M. Rehmani. Energy Replenishment using Renewable and Traditional Energy Resources for Sustainable Wireless Sensor Networks: A Review. *Renewable and Sustainable Energy Reviews, Vol*, 45:769–784, doi: 10.1016/j.rser.2015.02.021, May 2015.

[4] E. Ancillotti, R. Bruno, and M. Conti. The Role of Communication Systems in Smart Grids: Architectures, Technical Solutions and Research Challenges. *Computer Communications*, 36(17-18):1665–1697, doi: 10.1016/j.comcom.2013.09.004, November 2013.

[5] Arbiter Systems. Model 1133a Power Sentinel - GPS-Synchronized Power Quality/Revenue Standard. Product Manual, Arbiter Systems, Inc., Paso Robles, 2016.

[6] Arbiter Systems. Model 1133a Power Sentinel - Synchronized Power Measurement Series. Product Manual, Arbiter Systems, Inc., Paso Robles, 2016.

[7] M. Audeh. Metropolitan-scale Wi-Fi mesh networks. *Computer*, 37(12):119–121, doi: 10.1109/MC.2004.251, December 2004.

[8] Australian Signals Directorate (ASD): Cyber Security Operations Center. Strategies to Mitigate Targeted Cyber Intrusions. Attack Mitigation Strategy, Australian Government: Department of Defence, February 2014.

[9] AV-TEST Threat Research Team. Locky/ Dridex/ Cryptolocker Analysis. White Paper, AV-TEST, February 2016.

[10] J. Barcelo-Ordinas, C. Bonnet, and F. Filali. OLSR and MPR: Mutual Dependences and Performances. Research Report RR-05-138, March 2005.

[11] P. Barford, M. Dacier, T. Dietterich, and M. Frederikson. Cyber SA: Situational Awareness for Cyber Defense. In *Cyber Situational Awareness*, volume 46, pages 3–13. Springer, 2010.

[12] BDEW German Association of Energy and Water Industries. Smart Grids Traffic Light Concept. Technical Report Smart Grid Traffic Light Concept, German Association of Energy and Water Industries (BDEW), Germany, March 2015.

[13] B. Bencsáth, G. Pék, L. Buttyán, and M. Félegyházi. The Cousins of Stuxnet: Duqu, Flame, and Gauss. *Future Internet*, 4(4):971–1003, doi: 0.3390/fi4040971, September 2012.

[14] R. Blom, M. Korman, R. Lagerström, and M. Ekstedt. Analyzing Attack Resilience of an Advanced Meter Infrastructure Reference Model. In *2016 Joint Workshop on Cyber- Physical Security and Resilience in Smart Grids (CPSR-SG)*, pages 1–6, doi: 10.1109/CPSRSG.2016.7684095, April 2016.

[15] Board on Energy and Environmental Systems; Division on Engineering and Physical Sciences; National Research Council. *Terrorism and the Electric Power Delivery System.* National Academies Press, ISBN: 978-0-309-11404-2, Washington, D.C., October 2012.

[16] Board on Energy and Environmental Systems; Division on Engineering and Physical Sciences; National Research Council. *The Resilience of the Electric Power Delivery System in Response to Terrorism and Natural Disasters: Summary of a Workshop.* National Academies Press, ISBN: 978-0-309-29395-2, Washington, D.C., October 2013.

[17] E. Bou-Harb, C. Fachkha, M. Pourzandi, M. Debbabi, and C. Assi. Communication Security for Smart Grid Distribution Networks. *IEEE Communications Magazine*, 51(1):42–49, doi: 10.1109/MCOM.2013.6400437, January 2013.

[18] C. Bruno, L. Guidi, A. Lorite-Espejo, and D. Pestonesi. Assessing a Potential Cyberattack on the Italian Electric System. *IEEE Security and Privacy*, 13(5):42–51, doi: 10.1109/MSP.2015.99, September 2015.

[19] G. Burke and J. Fahey. AP Investigation: US Power Grid Vulnerable to Foreign Hacks. [Online] Available: http://bigstory.ap.org/article/c8d531ec05e0403a90e9d3ec0b8f83c2/ap-investigation-us-power-grid-vulnerable-foreign-hacks, December 2015.

[20] M. Burnett and D. Kleiman. *Perfect Passwords*, volume 1, ISBN: 978-1-59749-041-2. Syngress, Burlington, 2005.

[21] CEN-CENELEC-ETSI Smart Grid Coordination Group. Smart Grid Information Security - Reference Architecture. Technical Report SG -CG/M490/I, European Committee for Standardization (CEN) - European Committee for Electrotechnical Standardization (CENELEC) - European Telecommunications Standards Institute (ETSI), December 2014.

[22] Centers for Disease Control and Prevention. Principles of Epidemiology. [Online] Available: http://www.cdc.gov/ophss/csels/dsepd/SS1978/Lesson1/Section11.html, May 2016.

[23] M. K. Chavan and P. V. Madane. Modelling and Detection of Camouflaging Worms - A Survey. *International Journal of Emerging Technology and Advanced Engineering*, 2(10):564–569, ISSN: 2250 – 2459, October 2012.

[24] T. Chown. IPv6 Implications for Network Scanning. Request for Comments RFC5157, IETF, March 2008.

[25] G. Christiner. Die Rolle der APG für die Stromversorgungssicherheit - Nationale und Internationale Herausforderungen. Technical Report 20903, E-Control, Austria, May 2013.

[26] D. Cimpean, P. Cano, F. García, K. Moulinos, and T. Haeberlen. Appropriate Security Measures for Smart Grids. Technical report, European Network and Information Security Agency (ENISA), December 2012.

[27] Communications Security, Reliability and Interoperability Council (CSRIC). Legacy Systems Risk Reductions. Technical Report CSRIC5-WG10, Federal Communications Commission (FCC), March 2017.

[28] A. Conta and M. Gupta. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. Request for Comments RFC4443, March 2006.

[29] P. J. Criscuolo. Distributed Denial of Service: Trin00, Tribe Flood Network, Tribe Flood Network 2000, and Stacheldraht ciac-2319. Technical Report UCRL-ID-136939 Rev. 1, Lawrence Livermore National Laboratory, February 2000.

[30] A. Dabrowski, J. Ullrich, and E. R. Weippl. Grid Shock: Coordinated Load-Changing Attacks on Power Grids: The Non-Smart Power Grid is Vulnerable to Cyber Attacks as Well. pages 303–314, doi: 10.1145/3134600.3134639. ACM Press, 2017.

[31] A. Dainotti, A. King, K. Claffy, F. Papale, and A. Pescape. Analysis of a "/0" Stealth Scan From a Botnet. *IEEE/ACM Transactions on Networking*, 23(2):341–354, doi: 10.1109/TNET.2013.2297678, April 2015.

[32] L. Dan and H. Bo. Advanced Metering Standard Infrastructure for Smart Grid. In *2012 China International Conference on Electricity Distribution (CICED)*, pages 1–4, doi: 10.1109/CICED.2012.6508429, September 2012.

[33] E. Davies. Recommendations for Filtering ICMPv6 Messages in Firewalls. Request for Comments RFC4890, IETF, May 2007.

[34] Dragos Inc. CRASHOVERRIDE Analyzing the Threat to Electric Grid Operations. Technical Report version 2.20170613, June 2017.

[35] B. Droste-Franke, B. Paal, C. Rehtanz, D. U. Sauer, J. P. Schneider, and M. Schreurs. *Balancing Renewable Electricity: Energy Storage, Demand Side Management, and Network Extension from an Interdisciplinary Perspective*, volume 40, ISBN: 978-3-642-25157-3. Springer, Germany, January 2012.

[36] P. Eder-Neuhauser. Technischer Vergleich von Maßnahmen zur Erhöhung der Hosting Capacity in ländlichen Verteilnetzen. Master's thesis, University of Applied Sciences Technikum Wien, Austria, September 2013.

[37] P. Eder-Neuhauser. Sicherheit in Informations und Kommunikationstechnologien für Smart Grids. In *eNNOVATION 2015*, ENERGYbase: Technikum Wien, Austria, December 2015.

[38] P. Eder-Neuhauser. Aktuelle Gefahrenlage: Überblick über die letzten Angriffe und Gefahrenpotential für Ihr EVU. In *WARNUNG - Unbefugter Zugriff auf Ihr System!*, Twin Conference Center, Austria, December 2016.

[39] P. Eder-Neuhauser. Gefahren für kritische Infrastrukturen durch Vernetzung. In *ENISA European Cyber Security Month - Security Potpourri*, Technikum Wien, Austria, October 2016.

[40] P. Eder-Neuhauser. Kritische IT-Infrastrukturen durch Vernetzung: Wie real ist die Angst vor dem großen Blackout. In *EPCON 2016*, Mauerbach, Austria, April 2016.

[41] P. Eder-Neuhauser. Defense measures for smart grid operators. [Online]. Available: https://www.cybersecurityaustria.at/index.php/blog/2016/457-defense-measures-for-smart-grid-operators, October 2017.

[42] P. Eder-Neuhauser. Kritische Infrastrukturen und Ihre Abhängigkeiten. Poster Session at Cybersecurityaustria, Austria. [Online] Available: https://www.cybersecurityaustria.at/index.php/home/kritische-infrastrukturen, January 2017.

[43] P. Eder-Neuhauser and T. Zseby. The Art of Defending Critical Infrastructures. Turin, Italy, September 2017. ISGT-Europe, IEEE Conference.

[44] P. Eder-Neuhauser, T. Zseby, and J. Fabini. Resilience and Security: A Qualitative Survey of Urban Smart Grid Architectures. *IEEE Access*, 4:839–848, doi: 10.1109/ACCESS.2016.2531279, February 2016.

[45] P. Eder-Neuhauser, T. Zseby, and J. Fabini. Simulations on Resilience and Malware Containment in Smart Grid Communication Architectures. In *VSS - VIENNA young SCIENTISTS SYMPOSIUM, June 9-10 2016*, pages 88–89, ISBN: 978–3–9504017–2–1, Gumpoldskirchen, Austria, June 2016. Book-of-Abstracts.com, Heinz A. Krebs.

[46] P. Eder-Neuhauser, T. Zseby, and J. Fabini. Malware Propagation in Smart Grid Networks: Simulation and Comparison of Three Malware Types. *submitted to: Journal of Computer Virology and Hacking Techniques*, 2017.

[47] P. Eder-Neuhauser, T. Zseby, and J. Fabini. Malware Propagation in Smart Grid Monocultures. *submitted to: e & i Elektrotechnik und Informationstechnik*, February 2018.

[48] P. Eder-Neuhauser, T. Zseby, J. Fabini, and G. Vormayr. Cyber Attack Models for Smart Grid Environments. *Sustainable Energy, Grids and Networks*, (12C (2017)):10–29, doi: 10.1016/j.segan.2017.08.002, August 2017.

[49] E. Egozcue, D. Rodríguez, J. Ortiz, V. Villar, and L. Tarrafeta. Smart Grid Security - Recommendations for Europe and Member States. Technical Report Deliverable:2012-07-01, European Network and Information Security Agency (ENISA), July 2012.

[50] Energy Institute Johannes Kepler University. Blackout Simulator, Jul 2016. [Online]. Available: http://www.blackout-simulator.com/.

[51] European Commission. Green Paper on a European Programme for Critical Infrastructure Protection. Technical Report COM(2005) 576 final, Commission of the European Communities, European Union, November 2005.

[52] European Network of Transmission System Operators (ENTSO-E). Member Companies. [Online] Available: www.entsoe.eu/about-entso-e/inside-entso-e/member-companies, July 2015.

[53] European Network of Transmission System Operators (ENTSO-E). Network Code on Emergency and Restoration (ER). Technical Report 150325_ENTSO-E_NC ER, European Network of Transmission System Operators (ENTSO-E), Belgium, May 2015.

[54] European Telecommunications Standards Institute (ETSI). Digital Cellular Telecommunications System (Phase 2+); Physical Layer on the Radio Path; General Description (3gpp TS 45.001 Version 11.0.0 Release 11). Technical Report TS 145 001 - V11.0.0 (2012-10), European Telecommunications Standards Institute (ETSI), Sophia Antipolis Cedex - FRANCE, 2012.

[55] European Telecommunications Standards Institute (ETSI). TC SES Satellite Earth Solutions. Technical Report ETSI TR 102 641, European Telecommunications Standards Institute (ETSI), 2012.

[56] F-Secure Labs. Blackenergy & Quedagh - The Convergence of Crimeware and APT Attacks. White Paper No. 1030745, F-Secure Labs, Helsinki, September 2014.

[57] F-Secure Labs. CozyDuke. White Paper No. 1030745, F-Secure Labs, Helsinki, 2015.

[58] N. Falliere, L. O. Murchu, and E. Chien. W32. Stuxnet Dossier. White Paper, Symantec Security Response Vol. 05, February 2011.

[59] J. Faulhaber, D. Felstead, H. Henry, and J. Jones. Microsoft Security Intelligence Report - Zeroing in on Malware Propagation Methods. [Online] Available: http://download.microsoft.com/download/0/3/3/0331766e-3fc4-44e5-b1ca-2bdeb58211b8/microsoft_security_intelligence_report_volume_11_zeroing_in_on_ malware_propagation_methods_english.pdf, June 2011.

[60] Federal Ministry of the Interior. National Strategy for Critical Infrastructure Protection (CIP Strategy). Technical Report 34423, Federal Ministry of the Interior, Germany, July 2009.

[61] Federal Office for Information Security (BSI). Protection Profile for the Security Module of a Smart Metering System (Security Module PP). Common Criteria Protection Profile BSI-CC-PP-0077-2013, Germany, December 2013.

[62] Federal Office for Information Security (BSI). Protection Profile for the Security Module of a Smart Meter Gateway (Security Module PP). Common Criteria Protection Profile BSI-CC-PP-0077-V2-2015, Germany, January 2015.

[63] Federal Office of Civil Protection and Distaster Assistance. Certification Report BSI-CC-PP-0073: Protection Profile for the Gateway of a Smart Metering System. White Paper BSI-CC-PP-0073, Federal Office for Information Security (BSI), Germany, March 2014.

[64] Federal Office of Civil Protection and Distaster Assistance. Kritis - Sector: Energy. White Paper [Online] Available: https://www.kritis.bund.de/SubSites/Kritis/EN/introduction/sectors/energy/energy_node.html, Germany, February 2015.

[65] N. Freed. Behavior of and Requirements for Internet Firewalls. Request for Comments RFC2979, IETF, October 2000.

[66] S. Gold. Cracking GSM. *Network Security*, 2011(4):12–15, doi: 10.1016/S1353–4858(11)70039–3, April 2011.

[67] D. Gollmann, P. Gurikov, A. Isakov, M. Krotofil, J. Larsen, and A. Winnicki. Cyber-Physical Systems Security: Experimental Analysis of a Vinyl Acetate Monomer Plant. In *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security*, CPSS '15, pages 1–12, doi: 10.1145/2732198.2732208, New York, NY, USA, April 2015. ACM.

[68] V. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. Hancke. A survey on smart grid potential applications and communication requirements. *IEEE Transactions on Industrial Informatics*, 9(1):28–42, doi: 10.1109/TII.2012.2218253, 2013.

[69] J. Göllner, L. Langer, and M. Tischlinger. *Smart Grid Security Guidance - (SG2)*. Schriftenreihe der Landesverteidigungsakademie. ISBN: 978-3-902944-98-6. Bundesministerium für Landesverteidigung und Sport (BMLVS), May 2016.

[70] HBGary, Inc. Operation Aurora: Detect, Diagnose, Respond. [Online] Available: http://paper.seebug.org/papers/APT/APT_cybercriminal_campagin/2010/Aurora_hbgary_draft.pdf, January 2010.

[71] J. Homan, S. MCBride, and R. Caldwell. IRONGATE ICS Malware: Nothing to See Here... Masking Malicious Activity on SCADA Systems - FireEye Threat Research Blog. [Online] Available: https://www.fireeye.com/blog/threat-research/2016/06/irongate_ics_malware.html, July 2016.

[72] Idiom. Binaryforest - AlienSpy Java Rat Overview. [Online] Available: http://blog.idiom.ca/2015/03/alienspy-java-rat-overview.html, 2016.

[73] IEC. Smart Grid Standards Map. [Online] Available: http://smartgrid standardsmap.com/, May 2016.

[74] IEEE Std 1901.2-2013. IEEE Standard for Low-Frequency (less than 500 kHz) Narrowband Power Line Communications for Smart Grid Applications. Technical Report 10.1109/IEEESTD.2013.6679210, December 2013.

[75] IEEE Std 802.11ai-2016 (Amendment to IEEE Std 802.11-2016). IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Fast Initial Link Setup. pages 1–164. doi: 10.1109/IEEESTD.2016.7792308, December 2016.

[76] IEEE Std C37.118.1-2014. *IEEE Standard for synchrophasor measurements for power systems. Amendment 1, Amendment 1: C37.118.1-2014. ISBN: 978-0-7381-8978-9.* 2014.

[77] V. Igure, S. Laughter, and R. Williams. Security Issues in SCADA Networks. *Computers & Security*, 25(7):498–506. doi: 10.1016/j.cose.2006.03.001, October 2006.

[78] A. Ilo. The Energy Supply Chain Net. *Energy and Power Engineering*, 05(05):384–390. doi: 10.4236/epe.2013.55040, June 2013.

[79] ISO/IEC Std 14908-3:2012. Information Technology - Control Network Protocol - Part 3: Power Line Channel Specification. Technical Standard, February 2012.

[80] ISO/IEC Std 27001:2013. Information Technology - Security Techniques - Information Security Management Systems - Requirements. Technical report, IT Governance Publishing, October 2013.

[81] ISO/IEC Std 27002:2013. Information Technology - Security Techniques - Code of Practice for Information Security Controls (Second Edition). Technical report, IT Governance Publishing, October 2013.

[82] ISO/IEC Std 7498-1:1994. Information technology – Open Systems Interconnection – Basic Reference Model. International Standard, November 1994.

[83] ISO/IEC Std 7498-1:1994. Open Systems Interconnection – Basic Reference Model. Technical report, November 1994.

[84] ISO/IEC TR 27019:2013. Information Technology - Security Techniques. Technical report, July 2013.

[85] P. Jacquet, A. Laouiti, P. Minet, and L. Viennot. Performance Analysis of OLSR Multipoint Relay Flooding in Two Ad Hoc Wireless Network Models. *No. 4260*, 1(L'archive ouverte pluridisciplinaire):26. doi: inria–00072327, May 2006.

[86] X. Jiang, J. Zhang, B. Harding, J. Makela, and A. Dominguez-Garcia. Spoofing GPS Receiver Clock Offset of Phasor Measurement Units. *IEEE Transactions on Power Systems*, 28(3):3253–3262. doi: 10.1109/TPWRS.2013.2240706, August 2013.

[87] M. Kaczmarek. Malware Instrumentation Application to Regin Analysis. *Le journal de la cybercriminalité & des investigations numériques*, 1(1):1–12. doi: http://dx.doi.org/10.18464/cybin.v1i1.2, December 2015.

[88] M. Kalenderi, D. Pnevmatikatos, I. Papaefstathiou, and C. Manifavas. Breaking the GSM A5/1 cryptography algorithm with rainbow tables and high-end FPGAS. In *22nd International Conference on Field Programmable Logic and Applications (FPL)*, pages 747–753. doi: 10.1109/FPL.2012.6339146, August 2012.

[89] V. Kamluk and A. Gostev. Adwind - A Cross Plattform RAT. White Paper V. 3.0 #Adwind, Kaspersky Labs, February 2016.

[90] M. Kammerstetter, L. Langer, F. Skopik, and W. Kastner. Architecture-driven Smart Grid Security Management. In *Proceedings of the 2nd ACM Workshop on Information Hiding and Multimedia Security*, pages 153–158. doi: 10.1145/2600918.2600937, New York, NY, USA, June 2014. ACM.

[91] kamstrup. Radio Mesh Network - High performance communications network for smart meters. [Online] Available: http://products.kamstrup.com/ajax/downloadFile.php?uid=5489748a8f820&pid=384, December 2015.

[92] kamstrup. The intelligent Network RF Concentrator - Data sheet http://products.kamstrup.com. [Online] Available: /ajax/downloadFile.php?uid=5360cd87b8d2f&display=1, May 2016.

[93] M. Kanabar, M.G. Adamiak, and J. Rodrigues. Optimizing Wide Area Measurement System Architectures with Advancements in Phasor Data Concentrators (PDCs). In *2013 IEEE Power and Energy Society General Meeting (PES)*, pages 1–5. doi: 10.1109/PESMG.2013.6672987, July 2013.

[94] Kaspersky Labs. The Regin Platform: Nation-State Ownage of GSM Networks. White Paper, Moscow, November 2014.

[95] Kaspersky Labs. Equation Group: Questions and Answers. Technical Report Report No. 1.5 #EquationAPT, Moscow, February 2015.

[96] Kaspersky Labs. The Mystery of Duqu 2.0 a Sophisticated Cyberespionage Actor returns. White Paper, Moscow, June 2015.

[97] Kaspersky Labs. The Flame: Questions and Answers. [Online] Available: https://securelist.com/blog/incidents/34344/the-flame-questions-and-answers-51/, May 2016.

[98] Kaspersky Labs. Gauss: Nation-State Cyber-Surveillance meets Banking Trojan. [Online] Available: https://securelist.com/blog/incidents/33854/gauss-nation-state-cyber-surveillance-meets-banking-trojan-54/, May 2016.

[99] Kaspersky Labs. Targeted Cyberattacks Logbook. [Online] Available: https://apt.securelist.com/#secondPage, May 2016.

[100] T. Kaufmann, D. Bothe, W. Gawlik, and K. Ponweiser. Optimization of Load Flows in Urban Hybrid Networks. In A. Bisello, Daniele Vettorato, Richard Stephens, and Pietro Elisei, editors, *Smart and Sustainable Planning for Cities and Regions*, Green Energy and Technology, pages 3–13. doi: 10.1007/978–3–319–44899–2_1. Springer International Publishing, 2017.

[101] T. Kaufmann, W. Gawlik, J. Marchgraber, and M. Litzlbauer. Islanding capabilities and requirements of grids and ICT structures. In *Conference on Energy Informatics 2014, 3rd D-A-CH*, Zurich, Switzerland, July 2014.

[102] G. Kerber and R. Witzmann. Statistische Analyse von NS-Verteilungsnetzen und Modellierung von Referenznetzen. *ew Fachthema Netze*, 6(6222):22–26, 2008.

[103] A.A. Khan, M.H. Rehmani, and M. Reisslein. Cognitive Radio for Smart Grids: Survey of Architectures, Spectrum Sensing Mechanisms, and Networking Protocols. *IEEE Communications Surveys Tutorials*, 18(1):860–898. doi: 10.1109/COMST.2015.2481722, March 2015.

[104] R. Khan, K. McLaughlin, D. Laverty, and S. Sezer. Threat Analysis of BlackEnergy Malware for Synchrophasor based Real-time Control and Monitoring in Smart Grid. pages 53–63. doi: 10.14236/ewic/ICS2016.7, Queen's Belfast University, UK, August 2016.

[105] J. Klick, S. Lau, D. Marzin, J. Malchow, and V. Roth. Internet-facing PLCs as a Network Backdoor. In *2015 IEEE Conference on Communications and Network Security (CNS)*, pages 524–532. doi: 10.1109/CNS.2015.7346865, September 2015.

[106] P. Kocher, D. Genkin, D. Gruss, W. Haas, and M. Hamburg. Spectre Attacks: Exploiting Speculative Execution. *arXiv preprint arXiv:1801.01203*, January 2018.

[107] M. Korki, H. Vu, C. Foh, X. Lu, and N. Hosseinzadeh. MAC performance evaluation in low voltage PLC networks. *ENERGY*, pages 135–140, 2011.

[108] E. Kovacs. PLC-Blaster Can Pose A Serious Threat to Industrial Networks, SecurityWeek. [Online] Available: http://www.securityweek.com/plc-worms-can-pose-serious-threat-industrial-networks, May 2016.

[109] K. Kruglov. Security Policies: Misuse of Resources, Kaspersky Labs. [Online] Available: https://securelist.com/blog/security-policies/35945/security-policies-misuse-of-resources/, May 2016.

[110] F. Kupzog. Self-Controlled Exploitation of Energy Cost Saving Potentials by Implementing Distributed Demand Side Management. In *2006 IEEE International Conference on Industrial Informatics*, pages 375–380. doi: 10.1109/INDIN.2006.275829, August 2006.

[111] L. T. Berger and K. Iniewski. *Wiley: Smart Grid Applications, Communications, and Security. ISBN: 978-1-118-00439-5.* April 2012.

[112] L Langer and M. Kammerstetter. SG2 - Smart Grid Security Guidance. Technical Report SG2_Poster_SGW2014, Austrian Institute of Technology, Austria, October 2014.

[113] R. Lee, M. Assante, and T. Conway. Analysis of the Cyber Attack on the Ukrainian Power Grid. White Paper, Defense Use Case, E-ISAC Electricity Sector Information Sharing & Analysis Center, March 2016.

[114] Pele Li, M. Salour, and Xiao Su. A Survey of Internet Worm Detection and Containment. *Communication Surveys and Tutorials*, 10(1):20–35. doi: 10.1109/COMST.2008.4483668, April 2008.

[115] M. Line, A. Zand, G. Stringhini, and R. Kemmerer. Targeted Attacks against Industrial Control Systems: Is the Power Industry Prepared? In *SEGS '14 Proceedings of the 2nd Workshop on Smart Energy Grid Security*, pages 13–22. doi: 10.1145/2667190.2667192, Arizona, US, November 2014. ACM Press.

[116] M. Lipp, M. Schwarz, D. Gruss, T. Prescher, and W. Haas. Meltdown. *arXiv preprint arXiv:1801.01207*, January 2018.

[117] Chun-Hao Lo and Nirwan Ansari. The Progressive Smart Grid System from Both Power and Communications Aspects. *IEEE Communications Surveys & Tutorials*, 14(3):799–821. doi: 10.1109/SURV.2011.072811.00089, 2011.

[118] Lockheed Martin Corporation. Seven Ways to Apply the Cyber Kill Chain with a Threat Intelligence Platform. White Paper PIRA# CMK201507003, 2015.

[119] I. C. Lu, Chia-Chien Wei, Hsing-Yu Chen, Kuan-Zhou Chen, Cheng-Hsiang Huang, Kai-Lun Chi, J. W. Shi, Fan-I. Lai, and Dan-Hua Hsieh. High-speed and duo-mode 850 nm VCSELs for 47 Gbps optical interconnect over 1 km OM4 fiber. In *2015 Optical Fiber Communications Conference and Exhibition (OFC)*, pages 1–3. doi: 10.1364/OFC.2015.W1D.3, March 2015.

[120] Wenpeng Luan, D. Sharp, and S. Lancashire. Smart grid communication network capacity planning for power utilities. In *IEEE PES T D 2010*, pages 1–4. doi: 10.1109/TDC.2010.5484223, April 2010.

[121] J. P. Macker and J. W. Dean. A Study of Link State Flooding Optimizations for Scalable Wireless Networks. Technical report, Naval Research Laboratory, Information Technology Division, 4555, 2003.

[122] L. Marinos. Smart Grid Threat Landscape and Good Practice Guide. White Paper, European Network and Information Security Agency (ENISA), December 2013.

[123] A. Matrosov, E. Rodionov, and D. Harley. Stuxnet under the Microscope. Final Report No. 1.31, eset, January 2011.

[124] R. Mattioli and K. Moulinos. Communication Network Interdependencies in Smart Grids. Technical Report 978-92-9204-139-7, European Network and Information Security Agency (ENISA), January 2016.

[125] R. Mattioli and K. Moulinos. Communication Network Interdependencies in Smart Grids - Annexes. Technical Report 978-92-9204-140-3, European Network and Information Security Agency (ENISA), January 2016.

[126] W. Mazurczyk and L. Caviglione. Information Hiding as a Challenge for Malware Detection. *IEEE Security Privacy*, 13(2):89–93. doi: 10.1109/MSP.2015.33, March 2015.

[127] J. McCarthy, O. Alexander, S. Edwards, D. Faatz, C. Peloquin, S. Symington, A. Thibault, J. Wiltberger, and K. Viani. Situational Awareness For Electric Utilities. Special Publication # 1800-7, National Institute of Standards and Technology, August 2010.

[128] Kieran McLaughlin (QUB). Smart Grid Protection Against Cyber Attacks (SPARKS): High Level Design Documentation and Deployment Architecture for Multi-Attribute SCADA Intrusion Detection Systems. Project Deliverable D4.1 Contract No. 608224, AIT Austrian Institute of Technology, August 2015.

[129] Mitre Corporation. Common Vulnerabilities and Exposures (CVE) - The Standard for Information Security Vulnerability Names. [Online] Available: http://cve.mitre.org/index.html, March 2016.

[130] T. Mizuno, H. Takara, A. Sano, and Y. Miyamoto. Dense Space-Division Multiplexed Transmission Systems Using Multi-Core and Multi-Mode Fiber. *Journal of Lightwave Technology*, 34(2):582–592. doi: 10.1109/JLT.2015.2482901, January 2016.

[131] D. Moore, C. Shannon, and J. Brown. Code-Red: A Case Study on the Spread and Victims of an Internet Worm. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 273–284, Marseille, France, November 2002. ACM SIGCOMM/USENIX Internet Measurement Workshop.

[132] National Cybersecurity and Communications Integration Center, Industrial Control Systems Cyber Emergency Response Team. ICS-CERT Monitor. White Paper, Department of Homeland Security, USA, October 2013.

[133] National Institute of Standards and Technology. National Vulnerability Database, Automating Vulnerability Management, Security Measures and Compliance Checking. [Online] Available: https://nvd.nist.gov/, March 2016.

[134] Y. Nativ. theZoo: A repository of LIVE malwares for your own joy and pleasure. [Available] Online: https://github.com/ytisf/theZoo, February 2018. original-date: 2014-01-09T18:55:35Z.

[135] J. Nazario. Blackenergy DDoS Bot Analysis. White Paper, Arbor Networks, October 2007.

[136] North American Synchro Phasor Initiative. NASPI Homepage. [Online] Available: https://www.naspi.org/, March 2015.

[137] NS-3 Consortium. ns-3 Consortium. [Online] Available: https://www.nsnam.org/, July 2016.

[138] G. Ollmann. Blocking Shodan. [Online] Available: https://ipv6.net/news/blocking-shodan/, January 2016.

[139] olsr.org. Open Link State Routing Protocol - man page. [Online] Available: http://www.olsr.org/docs/olsrd.conf.5.html, December 2004.

[140] Open Source Geospatial Foundation (OSGeo). The QGIS project. [Online] Available: http://qgis.org/en/site/, October 2016.

[141] P. Paganini. Virus Discovered at the Gundremmingen Nuclear Plant in Germany. [Online] Available: http://securityaffairs.co/wordpress/46708/security/virus-gundremmingen-nuclear-plant.html, April 2016.

[142] P. Palensky and F. Kupzog. Smart grids. *Annual Review of Environment and Resources*, 38:201–226. doi: 10.1146/annurev–environ–031312–102947, 2013.

[143] D. Palma and M. Curado. Inside-Out OLSR Scalability Analysis. In *Ad-Hoc, Mobile and Wireless Networks*, pages 354–359. Springer, Berlin, Heidelberg, September 2009. DOI: 10.1007/978-3-642-04383-3_27.

[144] J. Pescatore, E. Skoudis, J. Ullrich, and M. Assante. The Six Most Dangerous New Attack Techniques and What's Coming Next. In *RSA Conference 2015*, USA, April 2015.

[145] T. Petermann, H. Bradke, A. Lüllmann, M. Poetzsch, and U. Riehm, editors. *What Happens During a Blackout: Consequences of a Prolonged and Wide-Ranging Power Outage.* Number 33 in Studies of the Office of Technology Assessment at the German Bundestag. ed. sigma, Berlin, 2 edition, 2013. ISBN: 978-3-7322-9329-2.

[146] J. Reichl, M. Schmidthaler, and F. Schneider. The value of supply security: The costs of power outages to Austrian households, firms and the public sector. *Energy Economics*, 36:256–261. doi: 10.1016/j.eneco.2012.08.044, March 2013.

[147] G.F. Riley, M.L. Sharif, and W. Lee. Simulating Internet Worms. In *The IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, 2004. (MASCOTS 2004). Proceedings*, pages 268–274. doi: 10.1109/MASCOT.2004.1348281. IEEE, October 2004.

[148] M. Roesch. Snort: Lightweight Intrusion Detection for Networks. In *LISA '99: 13 th Systems Administration Conference*, volume 99, pages 229–238, Washington, Seattle, 1999.

[149] C. Roger. Port Scanning Techniques and the Defense Against Them. White Paper, SANS Institute, October 2001.

[150] M. Rosas-Casals, S. Valverde, and R. Sole. Topological Vulnerability of the European Power Grid Under Errors and Attacks. *International Journal of Bifurcation and Chaos*, 17(7):2465–2475. doi: 10.1142/S0218127407018531, July 2006.

[151] I. Saeed, A. Selamat, and A. Abuagoub. A Survey on Malware and Malware Detection Systems. *International Journal of Computer Applications*, 67(16):25–31. doi: 10.5120/11480–7108, April 2013.

[152] SANS Institute. IAD's Top 10 Information Assurance Mitigation Strategies. [Online] Available: https://www.sans.org/security-resources/IAD_top_10_info_assurance_mitigations.pdf, July 2013.

[153] SANS Institute, Digital Forensics and Incident Response Team. Threat Intelligence Consumption. Technical Report DFIR-Intel-v1-4-16, 2016.

[154] B. Schneier. Attack Trees: Modeling Security Threats. *Dr. Dobb's Journal*, 24(12), December 1999.

[155] J. Schultz. Watering Hole Attacks an Attractive Alternative to Spear Phishing, on Cisco Security Blogs. [Online] Available: http://blogs.cisco.com/security/watering-hole-attacks-an-attractive-alternative-to-spear-phishing, 2016.

[156] A. Sendin, I. Peña, and P. Angueira. Strategies for Power Line Communications Smart Metering Network Deployment. *Energies*, 7(4):2377–2420. doi: 10.3390/en7042377, April 2014.

[157] D. Shin, S. He, and J. Zhang. Robust and Cost-Effective Architecture Design for Smart Grid Communications: A Multi-Stage Middleware Deployment Approach. In *2014 Proceedings IEEE INFOCOM*, pages 2822–2830. doi: 10.1109/INFOCOM.2014.6848232, April 2014.

[158] S. Shin, G. Gu, N. Reddy, and C. Lee. A Large-Scale Empirical Study of Conficker. *IEEE Transactions on Information Forensics and Security*, 7(2):676–690. doi: 10.1109/TIFS.2011.2173486, April 2012.

[159] Shodan. IoT Search Engine. [Online] Available: https://www.shodan.io/, June 2016.

[160] Y. Shoukry, P. Martin, Y. Yona, S. Diggavi, and M. Srivastava. Attack Resilience and Recovery using Physical Challenge Response Authentication for Active Sensors Under Integrity Attacks. *arXiv:1605.02062 [cs]*, May 2016. arXiv: 1605.02062.

[161] J. Silva. Understanding Wireless Topologies for Smart Grid Applications. In *Implementing Interoperability: Advancing Smart Grid Standards, Architecture and Community*, volume 5, page 70, United States of America, December 2011. Grid - Interop Forum.

[162] F. Sinitsyn. Locky: The Encryptor Taking the World by Storm, Kaspersky Labs. [Online] Available: https://securelist.com/blog/research/74398/locky-the-encryptor-taking-the-world-by-storm/, May 2016.

[163] F. Skopik and L. Langer. Cyber Security Challenges in Heterogeneous ICT Infrastructures of Smart Grids. *Journal of Communications*, 8:463, August 2013.

[164] SONET.com. SONET/SDH Digital Hierarchy. [Online] Available: http://www.sonet.com/EDU/edu.htm, November 2016.

[165] R. Spenneberg, M. Brüggemann, and H. Schwartke. PLC-Blaster: A Worm Living Solely in the PLC. In *Black Hat Asia 2016*, page 16, Singapore, April 2016.

[166] S. Staniford, V. Paxson, and N. Weaver. How to 0wn the Internet in Your Spare Time. In *USENIX Security Symposium*, pages 149–167. ISBN: 1–931971–00–5, San Francisco, August 2002.

[167] Statistik Österreich. *Gebäude- und Wohnungszählung 2001.* Statistik Austria, Wien, 2004. OCLC: 58527040. ISBN: 3-902452-95-1.

[168] P. Suganthi and A. Tamilarasi. Performance of OLSR routing protocol under different route refresh intervals in ad hoc networks. *International Journal on Computer Science and Engineering*, 3(1):133–137. ISSN: 0975–3397, 2011.

[169] Symantec. Regin: Top-Tier Espionage Tool Enables Stealthy Surveillance. White Paper, Symantec, November 2014.

[170] P. Ször and P. Ferrie. Hunting for metamorphic. In *Virus Bulletin Conference*, pages 123–144. CiteSeerX, September 2001.

[171] K. Tan. How can attacker use ICMP for reconnaissance?, SANS - Information Security Resources. [Online] Available: https://www.sans.org/security-resources/idfaq/how-can-attacker-use-icmp-for-reconnaissance/3/13, April 2016.

[172] V. Targon. A Cost Analysis of Wireless Mesh Networks. In *2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 349–356. doi: 10.1109/WIOPT.2014.6850319, May 2014.

[173] The Laboratory of Cryptography and System Security (CrySyS). W32.Duqu - The Precursor to the next Stuxnet. White Paper, Symantec Security Response, November 2011.

[174] The Smart Grid Interoperability Panel – Smart Grid Cybersecurity Committee. Guidelines for Smart Grid Cybersecurity. Technical Report NIST IR 7628r1, National Institute of Standards and Technology, September 2014. doi: 10.6028/NIST.IR.7628r1.

[175] A. Thigpen and E. Chien. W32.Sality, Security Response, Symantec. [Online] Available: https://www.symantec.com/security_response/writeup.jsp?docid=2006-011714-3948-99, 2013.

[176] O. Thonnard, L. Bilge, G. O'Gorman, S. Kiernan, and M. Lee. Industrial Espionage and Targeted Attacks: Understanding the Characteristics of an Escalating Threat. In Davide Balzarotti, Salvatore J. Stolfo, and Marco Cova, editors, *Research in Attacks, Intrusions, and Defenses*, number 7462 in Lecture Notes in Computer Science, pages 64–85. doi: 10.1007/978–3–642–33338–5_4. Springer Berlin Heidelberg, September 2012.

[177] O. Trullols-Cruces, M. Fiore, and J.M. Barcelo-Ordinas. Understanding, Modeling and Taming Mobile Malware Epidemics in a Large-Scale Vehicular Network. In *14th International Symposium and Workshops on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–9. doi: 10.1109/WoWMoM.2013.6583402, Madrid, June 2013. IEEE.

[178] S. Tzu. *Art of War.* ISBN: 978-1-85326-305-7, Hertfordshire: Wordsworth Editions Ltd, December 1999.

[179] J. Ullrich. Targeted IPv6 Scans Using pool.ntp.org. [Online] Available: https://isc.sans.edu/forums/diary/Targeted+IPv6+Scans+Using+poolntporg/20681/, June 2016.

[180] Union for the Co-Ordination of Transmission of Electricity. Final Report on System Disturbance on 4 November 2006. Technical Report Final-Report-20070130, UTCE, Belgium, November 2006.

[181] J. Van de Vyver, G. Deconinck, and R. Belmans. The Need for a Distributed Algorithm for Control of the Electrical Power Infrastructure. In *2003 IEEE International Symposium on Computational Intelligence for Measurement Systems and Applications, 2003. CIMSA '03*, pages 211–215. doi: 10.1109/CIMSA.2003.1227229, July 2003.

[182] Verizon Inc. 2017 Data Breach Investigations Report, 10th Edition. Technical Report WP16943 04/17, Verizon Inc., May 2017.

[183] W. Wang, Q. Cai, Y. Sun, and H. He. Risk-Aware Attacks and Catastrophic Cascading Failures in U.S. Power Grid. In *2011 IEEE Global Telecommunications Conference (GLOBECOM 2011)*, pages 1–6. doi: 10.1109/GLOCOM.2011.6133788, December 2011.

[184] Wiener Netze GmbH. Geospacial Data for the Electricity Grid of Vienna [Data redacted], District 1040 of Vienna, October 2016.

[185] Y. Yan, Y. Qian, H. Sharif, and D. Tipper. A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges. *IEEE Communications Surveys Tutorials*, 15(1):5–20. doi: 10.1109/SURV.2012.021312.00034, March 2013.

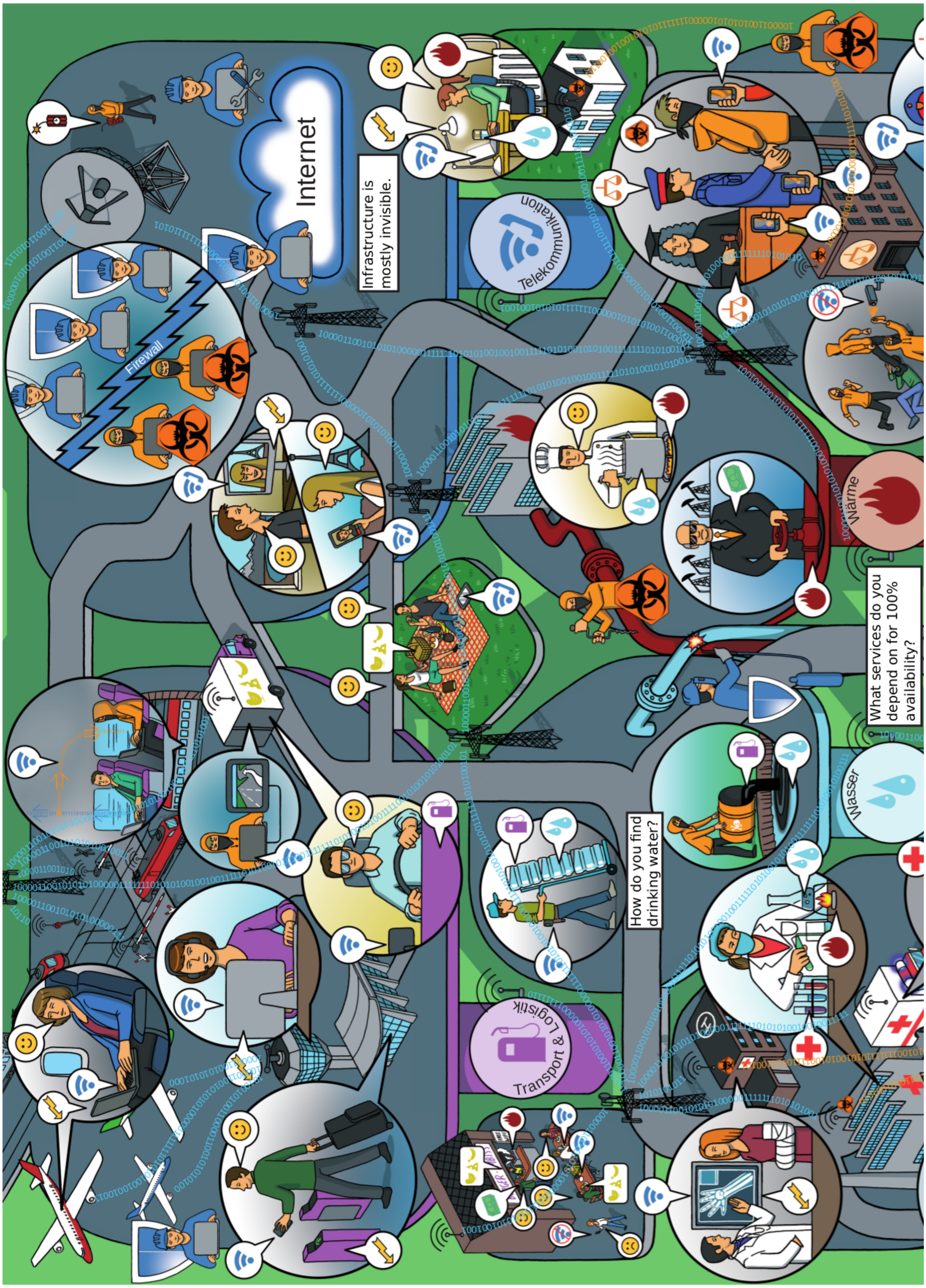
[186] R. Yu, Y. Zhang, S. Gjessing, C. Yuen, S. Xie, and M. Guizani. Cognitive Radio Based Hierarchical Communications Infrastructure for Smart Grid. *IEEE Network*, 25(5):6–14. doi: 10.1109/MNET.2011.6033030, September 2011.

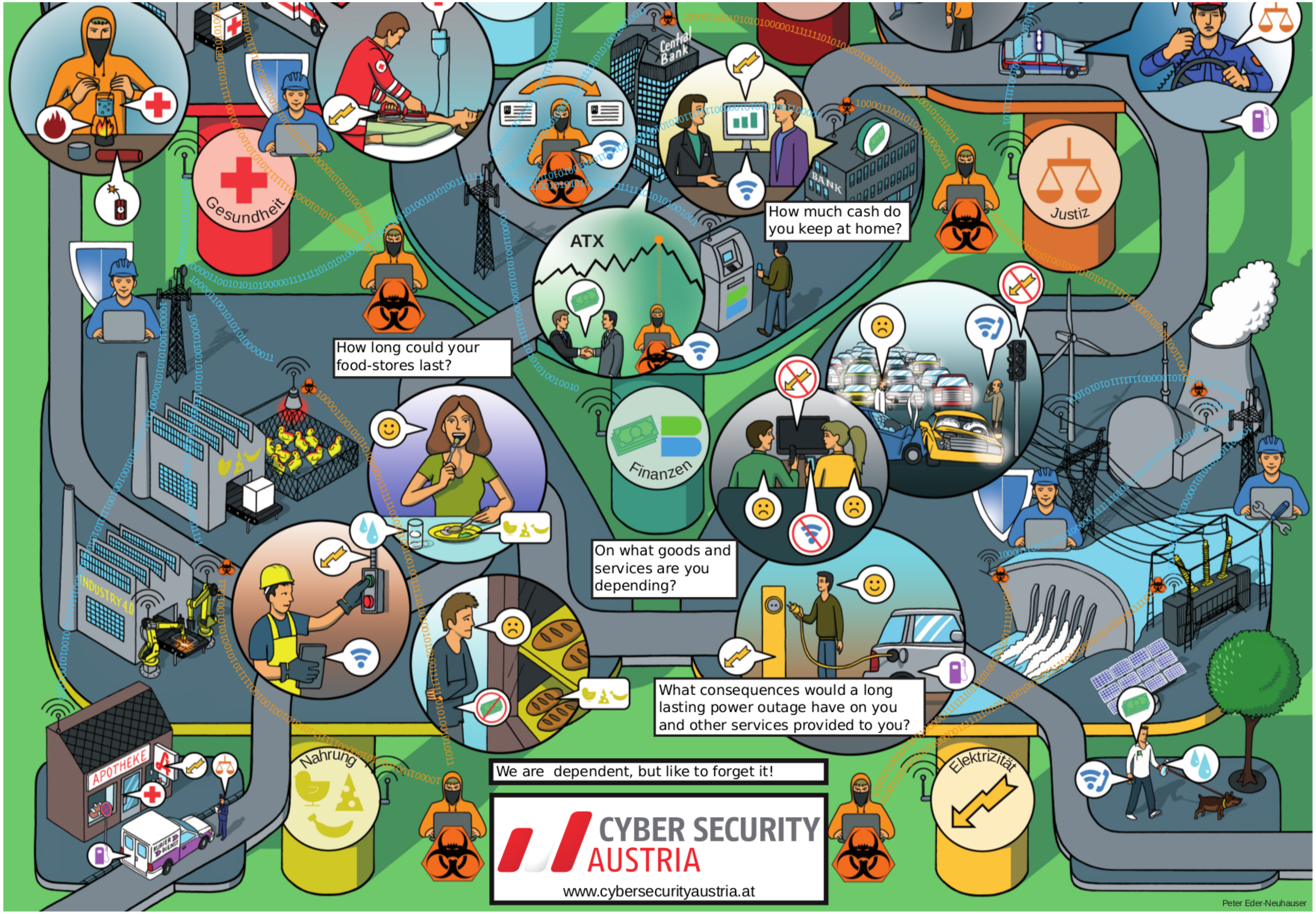
[187] W. Zhang and L. Yang. SC-FDMA for uplink smart meter transmission over low voltage power lines. In *2011 IEEE International Symposium on Power Line Communications and Its Applications*, pages 497–502. doi: 10.1109/ISPLC.2011.5764449, April 2011.

[188] X.-P. Zhang, C. Rehtanz, X. Bai, Z. Wu, and U. Hager. Towards European Smart Grids. In *2011 IEEE Power and Energy Society General Meeting*, pages 1–5. doi: 10.1109/PES.2011.6039028, July 2011.

[189] T. Zseby. Is IPv6 Ready for the Smart Grid? In *2012 International Conference on Cyber Security (CyberSecurity)*, pages 157–164. doi: 10.1109/CyberSecurity.2012.27, Washington, DC, December 2012. Fraunhofer Inst. for Open Commun. Syst. (FOKUS).

Appendix A: Overview of Critical Infrastructures, presented in [42]


Firewall
Internet
Infrastructure is mostly invisible.
Telekommunikation
Transport & Logistik
How do you find drinking water?
What services do you depend on for 100% availability?
Wasser
Wärme



How much cash do you keep at home?

How long could your food-stores last?

On what goods and services are you depending?

What consequences would a long lasting power outage have on you and other services provided to you?

We are dependent, but like to forget it!

CYBER SECURITY AUSTRIA

www.cybersecurityaustria.at

Peter Eder-Neuhauser

Appendix B: Related Standards and Books

This section lists a comprehensive set of standards, concepts and books, concerning smart grid security, as suggested in [15, 16, 20, 44, 49, 63, 64, 73, 76, 124, 125, 163]. The following sources should be considered for detailed implementations of the resilience measures introduced in Section 10.

Books

- M. Burnett and D. Kleiman. *Perfect Passwords*, volume 1, ISBN: 978-1-59749-041-2. Syngress, Burlington, 2005
- Board on Energy and Environmental Systems; Division on Engineering and Physical Sciences; National Research Council. *The Resilience of the Electric Power Delivery System in Response to Terrorism and Natural Disasters: Summary of a Workshop.* National Academies Press, ISBN: 978-0-309-29395-2, Washington, D.C., October 2013
- Board on Energy and Environmental Systems; Division on Engineering and Physical Sciences; National Research Council. *Terrorism and the Electric Power Delivery System.* National Academies Press, ISBN: 978-0-309-11404-2, Washington, D.C., October 2012

Standards and Concepts

- IEV IEC 61850-90-12 provides definitions, guidelines and recommendations for the engineering of WANs, especially regarding their protection, control and monitoring. It is based on IEC 61850 and several related protocol standards. It is mostly used for communications between substations and the control centere.
- IEC 61850: Substation Automation: Reference Architecture. Contains a number of relevant standards starting from IEC/TR 61850-1 through -10 and 80 through 90
- IEC 62351: is a standard designed to handle the security of several protocols including IEC 60870, IETC 61850, IEC 61970 and IEC 61968. Among its features, it includes TLS encryption, node authentication, message authentication and several other specific security profiles.
  - IEC 62351-3: Security for any profiles including TCP/IP, TLS, Authentication and Certificates
  - IEC 62351-4: Security for any profiles including Manufacturing Message Specification (MMS) (e.g., ICCP-based IEC 60870-6, IEC 61850, etc.), Authentication for MMS, TLS (RFC 2246) is

  inserted between RFC 1006 & RFC 793 to provide transport layer security

  - IEC 62351-5: Security for any profiles including IEC 60870-5 (e.g., Distributed Network Protocol (DNP3) derivative), TLS for TCP/IP profiles and encryption for serial profiles.
  - IEC 62351-6: Security for IEC 61850 profiles. VLAN's use is made as mandatory for GOOSE, RFC 2030 to be used for SNTP
  - IEC 62351-7: Security through network and system management. Defines Management Information Base (MIBs) that are specific for the power industry, to handle network and system management through SNMP based methods.
  - IEC 62351-8: Role-based access control. Covers the access control of users and automated agents to data objects in power systems by means of role-based access control (RBAC).
  - IEC 62351-9: Key Management: Describes the correct and safe usage of safety-critical parameters, e.g. passwords, encryption keys. Covers the whole life-cycle of cryptographic information (enrollment, creation, distribution, installation, usage, storage and removal). Methods for algorithms using asymmetric cryptography. Handling of digital certificates (public / private key). Setup of the Public Key Infrastructure (PKI) environment with X.509 certificates. Certificate enrollment by means of SCEP / CMP. Certificate revocation by means of CRL / OCSP. A secure distribution mechanism based on GDOI and the IKEv2 protocol is presented for the usage of symmetric keys, e.g. session keys.
  - IEC 62351-10: Security Architecture: Explanation of security architectures for the entire IT infrastructure. Identifying critical points of the communication architecture, e.g. substation control center, substation automation. Appropriate mechanisms security requirements, e.g. data encryption, user authentication. Applicability of well-proven standards from the IT domain, e.g. VPN tunnel, secure FTP, HTTPS.

- IEC 60870 Telecontrol (SCADA) and Communication Profiles for basic messages between 2 systems
  - IEC TS 60870-5-7: IEC 60870 defines SCADA. Ext -5-7 is a security extensions to IEC 60870-5-101 and IEC 60870-5-104 protocols (applying IEC 62351)
- IEC 61400-25: Data Transfer from Windpower generation to SCADA. Client-Server supported communication environment. The standard

addresses the issue of proprietary communication systems utilizing a wide variety of protocols.

- IEC 62056: Data Exchange Protocol for Smart Meter Data and Load control.
- IEC 62056-5-3: Specifies Companion Specification for Energy Metering (COSEM) application layer in terms of structure, services, protocols for client-server models. Defines services for establishing and releasing application associations and data communication services
- ISO 22301 Business Continuity Management: Understand and prioritize the threats to your business with the international standard for business continuity. ISO 22301 specifies the requirements for a management system to protect against, reduce the likelihood of and ensure your business recovers from disruptive incidents.
- ISO/IEC 27001: A systematic approach to managing sensitive company information so that it remains secure. It includes people, processes and IT systems by applying a risk management process.
- ISO/IEC 27002: Information technology, Security techniques, Code of practice for information security controls .
- ISO/IEC 27019: This Technical Report is intended to help organizations in “the energy industry” interpret and apply ISO/IEC 27002:2005 in order to secure their electronic process control systems.
- IEC 62443 series: Defines elements necessary to establish a cyber security management system (CSMS) for industrial automation and control systems.
- IEC 61158 Fieldbus Protocols for Realtime Control in distributed Systems
- NIST SP 800-82: SCADA systems, Distributed Control Systems (DCS), and other control system configurations such as PLCs.
- NIST SP 800-53: Guidelines for public institutions and outreach efforts in information system security, and on activity with industry, government, and academic organizations.
- BSI-CC-PP-0073 Federal Office of Civil Protection and Distaster Assistance: Protection Profile for the Gateway of a Smart Metering System,” Federal Office for Information Security (BSI), Germany, White Paper BSI-CC-PP-0073, Mar. 2014
- BSI-CC-PP-0077-2013 Federal Office for Information Security (BSI): BSI protection profile,” Federal Office for Information Security (BSI), Germany, BSI-CC-PP-0077-2013, 2013

- BSI-CC-PP-0077-V2-2015 Federal Office for Information Security (BSI): BSI protection profile," Federal Office for Information Security (BSI), Germany, BSI-CC-PP-0077-V2-2015, Jan. 2015
- North American Electric Reliability Corporation (NERC), "CIP Standards," Jun-2016. [Online]. Available: http://www.nerc.com/pa/Stand/Pages/CIPStandards.aspx. [Accessed: 02-Jun-2016].
- National Institute of Standards and Technology (NISTIR), "Guidelines for Smart Grid Cyber Security, document NISTIR 7628." National Institute of Standards and Technology, Aug-2010
- IEEE 1588 Network Time Sync in Precision Time Protocol (PTP)
- IEEE 1547 Connection Criteria for decentralized Systems to central control systems
- DNP3: Processautomation in SCADA
- DNP3 Secure: is an upgrade to the standard DNP3 protocol designed to provide additional security measures, including authentication and data encryption. It is compliant with IEC 62351-5 standard, and in some cases VPNs are also used to secure IP networks.
- Transport Layer Security (TLS): is a cryptographic protocol designed to protect communications over a network. TLS is using asymmetric cryptography and client certificates for authentication and symmetric cryptography for sessions
- Internet Protocol Security (IPSec): comprises of a set of protocols designed specifically to protect IP communications by applying authentication and encryption to them. It supports mutual authentication, and works on the Network Layer (while TLS works on the Application Layer)
- Secure Shell (SSH): is a protocol that provides a secure connection to remote machines, by applying encryption to protect the data
- Virtual Private Network (VPN): Not a protocol, but a concept. In a VPN the use of point-to-point private network over a public network or the Internet are specified. A VPN uses tunnelling protocols to make available private communication, and to also make use of encryption protocols to protect the confidentiality of the data transmitted.
- C37.118.1-2011 - IEEE Standard for Synchrophasor Measurements for Power Systems

Appendix C: Simulation Scenario Mapping to the Dataset

Table 38: Appendix C: Simulation-Scenarios mapping to the Dataset

Scenario name	Dataset	Simulation Model [1]
Maximum mesh-size, cf. Section 9.1.2	data_set/2017.04.03_test_mesh_limits	ENP_3_Mesh_no_border_nodes_old_Routing.cc [2]
Maximum overall network Size, cf. Section 9.1.2	data_set/2017.02.27_S1_proof_distance_has_no_effect_Extnd_to_160_Nodes	ENP_3_Mesh_no_border_nodes_old_Routing.cc [2]
Increased OLSR message cycle, cf. Section 9.1.3	data_set/Vergleich_OLSR5_vs_OLSR500	ENP_3_Mesh_no_border_nodes_old_Routing.cc [2]
Network segmentation and Monocultures, cf. section 9.2	data_set/2017.05.01_homogeneous_vs_heterogeneous_SEGMENTATION	ENP_3_Mesh_no_border_nodes_old_Routing.cc [2]
Centralized pandemic, cf. Section 9.3.1	data_set/1.centralized_topology	ENP_1_Centr_pandemic.cc
Centralized endemic, cf. Section 9.3.1	data_set/1.centralized_topology	ENP_1_Centr_endemic_contagion.cc
Centralized contagion, cf. Section 9.3.1	data_set/1.centralized_topology	ENP_1_Centr_endemic_contagion.cc
Cell topology pandemic, cf. Section 9.3.2	data_set/2.cell_topology	ENP_2.5_Cell_topol_pandemic.cc
Cell topology endemic, cf. Section 9.3.2	data_set/2.cell_topology	ENP_2_Cell_topol.cc
Cell topology Contagion, cf. Section 9.3.2	data_set/2.cell_topology	ENP_2_Cell_topol.cc
Mesh topology pandemic, cf. Section 9.3.3	data_set/3.mesh_topology	ENP_3_Mesh_no_border_nodes_old_Routing.cc
Mesh topology endemic, cf. Section 9.3.3	data_set/3.mesh_topology	ENP_3_Mesh_no_border_nodes_old_Routing.cc
Mesh topology contagion, cf. Section 9.3.3	data_set/3.mesh_topology	ENP_3_Mesh_no_border_nodes_old_Routing.cc
Decentralized pandemic, cf. Section 9.3.4	data_set/4.decentr_topology	ENP_4_decentr.cc
Decentralized endemic, cf. Section 9.3.4	data_set/4.decentr_topology	ENP_4_decentr.cc
Decentralized contagion, cf. Section 9.3.4	data_set/4.decentr_topology	ENP_4_decentr.cc

Notation:
(1) Corresponds to the simulation files in /auxiliary_files/ns3_models.
(2) Large mesh clusters are the most difficult test-case for simulation.