

Security Engineering and Software Development for Critical Infrastructure IT in the Cloud

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Software Engineering and Internet Computing

eingereicht von

Sarita Paudel

Matrikelnummer 1029083

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Priv.-Doz. Dr. Ivona Brandic
Mitwirkung: Dr. Markus Tauber (Austrian Institute of Technology-AIT, Vienna)

Wien, 14.07.2014

(Unterschrift Verfasserin)

(Unterschrift Betreuung)

Security Engineering and Software Development for Critical Infrastructure IT in the Cloud

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Software Engineering and Internet Computing

by

Sarita Paudel

Registration Number 1029083

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Priv.-Doz. Dr. Ivona Brandic

Assistance: Dr. Markus Tauber (Austrian Institute of Technology-AIT, Vienna)

Vienna, 14.07.2014

(Signature of Author)

(Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Sarita Paudel

Veichengasse 13-15/3, 2210 Gerasdorf bei Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)

(Unterschrift Verfasserin)

Abstract

With the increasing popularity of cloud computing, security in cloud-based applications is gaining awareness and security is regarded as one of the most crucial factors for the long-term success of such applications. Despite all the benefits of cloud computing, its fate lies in its success in gaining trust from its users that can be achieved only by ensuring safe and secure cloud environments. The objective of this research is to evaluate currently existing security standards and tools for Critical Infrastructure (CI) in cloud computing with focus on software development standards and tools to discuss their applicability in this context and to discuss how they support development of secure software and system engineering. A show case for a surveillance video or image storage system is used to experiment with a software development tool. We have identified security issues from literature review and experimentation with the show case. We have developed a multidimensional taxonomy based on the identified security issues and the existing standards and tools. The development of a multidimensional taxonomy is based on open security issues for CI in the Cloud, points out multiple standards and tools, and map security requirements to the available standards and tools. This multidimensional taxonomy will help software developers to identify appropriate means for creating secure cloud applications in CI domain.

Publications

As part of this work two articles have been published and are available in IEEE digital library, one position paper about motivational aspects for this work and other paper regarding the show case scenario focusing on multilayer cloud compliance considering technical and legal aspects. A Journal paper is currently in preparation.

- Sarita Paudel, Markus Tauber, Ivona Brandic. Security Standards Taxonomy for Cloud Applications in Critical Infrastructure IT. In: *The 8th International Conference for Internet Technology and Secured Transactions (ICITST-2013)*. IEEE; 2013.
- Markus Florian, Sarita Paudel, Markus Tauber. Trustworthy Evidence Gathering Mechanism for Multilayer Cloud Compliance. In: *The 8th International Conference for Internet Technology and Secured Transactions (ICITST-2013)*. IEEE; 2013.

Kurzfassung

Mit der steigenden Popularität von Cloud Computing, hat auch das Thema Informationssicherheit und Datenschutz einen hohen Stellenwert bekommen und wird in diesem Zusammenhang meist sehr heiß diskutiert. Der Erfolg von Cloud-basierten Diensten hängt davon ab, ob zu den Nutzern von Cloud-Angeboten das notwendige Vertrauen aufgebaut werden kann. Das wiederum kann nur dann gelingen, wenn zuverlässige und sichere Cloud Computing Umgebungen durch den Entwicklungsprozess und eingesetzten Technologien garantiert werden kann. Diese Forschungsarbeit hat zum Ziel gesetzt, existierende Sicherheitsstandards und Werkzeuge für Kritische Infrastrukturen (KI) in Cloud Umgebungen zu evaluieren, und deren Anwendbarkeit in der Softwareentwicklung abzuschätzen. Das Hauptaugenmerk bei der Evaluierung liegt auf dem Beitrag der Standards und Werkzeuge zur sicheren Software und System Engineering. Ein Anforderungskatalog für sicherheitsrelevante Anforderungen wurde einerseits durch die Literaturanalyse und andererseits mit einem Prototypen (Werkzeuge zur Speicherung von Bilder einer Überwachungskamera) erstellt. Basierend auf den Anforderungen, wurde eine mehrdimensionale Taxonomie erstellt, die die Beziehungen zwischen Anforderungen und existierenden Sicherheitsstandards und Werkzeugen darstellt. Diese Taxonomie hilft Softwareentwicklern dabei, die notwendigen Sicherheitsstandards zu identifizieren, wenn es darum geht sicherheitsrelevante Anforderungen für Cloud-basierte Infrastruktur zu realisieren.

Publikationen

Im Rahmen dieser Forschungsarbeit wurden bereits zwei wissenschaftliche Arbeiten veröffentlicht. Diese beiden Papers findet man in der IEEE Digital Library. An einem Journalpaper wird derzeit gearbeitet.

- Sarita Paudel, Markus Tauber, Ivona Brandic. Security Standards Taxonomy for Cloud Applications in Critical Infrastructure IT. In: *The 8th International Conference for Internet Technology and Secured Transactions (ICITST-2013)*. IEEE; 2013.
- Markus Florian, Sarita Paudel, Markus Tauber. Trustworthy Evidence Gathering Mechanism for Multilayer Cloud Compliance. In: *The 8th International Conference for Internet Technology and Secured Transactions (ICITST-2013)*. IEEE; 2013.

Acknowledgements

This research was conducted at the Austrian Institute of Technology in collaboration with the Vienna University of Technology and was supported by the SECCRIT (SEcure Cloud computing for CRITICAL infrastructure IT) project, which is a EU-funded research project in the 7th Framework Programme (FP7-SEC-2012-1). I owe my gratitude to all those people, who have made this thesis possible and because of whom my research experience has been the one that I will cherish forever.

My deepest gratitude goes to my supervisors Ivona Brandic and Markus Tauber. I have been amazingly fortunate to have two supervisors, who gave me the freedom to explore on my own and at the same time the guidance to recover when my steps faltered. I am heavily indebted to both my supervisors whose help, stimulating suggestions and encouragement helped me in all the time of research. Their patience and support also helped me publish two papers as part of this research – which would not have been possible without their advice.

AND of course, my family and friends have always been there, at all good and bad times to support me and my plans. My special thanks goes to my parents and parents-in-law, who made all this possible through decades of hard work. My husband Deepak has always supported me and his affectionate support has shaped my career more than anything else. My brothers and sisters Deepak, Suresh, Uma, Dinesh and Ratna have always been there to support and encourage me with their best wishes. Last but not least, I would like to thank our friend Guenter Zeh for his good wishes and encouraging words.

A large, elegant, cursive 'Thank You' graphic. The words 'Thank' and 'You' are written in a highly stylized, flowing script. The 'T' in 'Thank' is particularly large and ornate, with long, sweeping flourishes that extend across the word 'You'. The overall style is classic and decorative.

Contents

Contents	ix
1 Introduction and Motivation	1
1.1 Critical Infrastructure	1
1.2 Security Engineering	3
1.3 Research Open Issues	4
1.4 Research Method	5
1.5 Contributions	5
2 Background	7
2.1 Cloud Computing	7
2.2 Cloud Computing Types	8
2.3 Cloud Computing Service Models	9
2.4 Security	10
2.5 Design for security	12
2.6 Open Web Application Security Project	14
3 Related Work	15
3.1 Security Requirements and Assessment	15
3.2 Classification of Secure Development Means	17
3.3 Implementing software development standards	18
3.4 Commercial Products	19
3.5 Security Objectives in the Cloud	19
3.6 Security Threats in the Cloud	20
4 Research Methodology	23
4.1 Phases of Research	24
4.2 Literature Survey	24
4.3 Show Case Implementation	26
4.4 Mapping Study	26
4.5 Analysis based on a Selected Standard	26
4.6 Threats to Validity	27
5 Show Case and Security Issues Analysis	29

5.1	Show Case description	29
5.2	Technologies and Architecture	34
5.3	Computation Scenarios	36
5.4	Security Issues of CI in the Cloud and Anticipated Result	36
6	Categorization and Classification of Secure Software Development Means	45
6.1	Classification and Categorization	45
6.2	Sub-categorization	51
7	Mapping of Security Issues and Popular Secure Software Development and Security Engineering Means	55
7.1	Contribution of Security Means	55
7.2	Matrix of Supporting Means and Issues:	66
7.3	Summary Explanation of not Significant Security Means that do not support Security Issues	68
7.4	Multidimensional Taxonomy	69
8	Evaluation of Taxonomy	71
8.1	Evaluation Plan	71
8.2	Security Issues from Show Case	72
8.3	Selection of Security Issues to be evaluated	73
8.4	SDL Tool selection	74
8.5	Threat Modeling	75
8.6	SDL Threat Modeling Tool	76
8.7	Contribution of SDL Threat Modeling to address Security Issues	84
9	Summary and Conclusion	87
9.1	Summary	87
9.2	Conclusion	87
9.3	Future Work	88
A	Appendix	89
A.1	Standards	89
A.2	Applying SDL Threat Modeling Tool in Show Case	91
	Bibliography	97

Introduction and Motivation

With the increasing importance of securing Critical Infrastructure (CI) against growing and evolving cyber threats, the awareness for security issues and mitigation efforts against attempted disruptions of the CI is also getting mainstream. As the reliable functioning of Critical Infrastructure is crucial for the viability of public infrastructures, it is very important to closely analyse the processes and tools that are used for development of such CI solutions.

1.1 Critical Infrastructure

Critical Infrastructures are generally thought of as the key systems, services and functions whose disruption or destruction would have a debilitating impact on public health and safety, commerce, and national security or any combination of those matters. CI provide the utilities and basic needs for our modern life. These CIs include electricity, water supply, the Internet, and in so called smart-cities even traffic control and public safety CCTV systems¹. The increasing flexibility and unpredictable usage patterns of such utilities often means that many challenges such as load balancing can occur the utility networks we use, i.e. the Critical Infrastructures (CIs). The usage of modern IT systems to control and manage CI allows to deal with such issues. However, this exposes CI to cyber risks and results in demand for protection against cyber attacks, even more than traditional IT systems as failing CI may have a cascading effect on each other and hence fatal effects [1]. Devices for management and control IT for CI are normally equipped with limited computation resources. Also data from an individual device does not make sense when looked in isolation and hence data from multiple devices needs to be accumulated. Thus, adoption of cloud technologies allows CI to benefit from dynamic resources allocation for managing unpredictable load peaks.

As depicted in Figure 1.1, it is important to differentiate between cyber security issues in „non-essential“ software and services and applications in CI. Although domains such as energy,

¹Related issues are investigated in SEcure Cloud computing for CRITICAL infrastructure IT (SECCRIT) which is a multidisciplinary research project. For more information see www.seccrit.eu

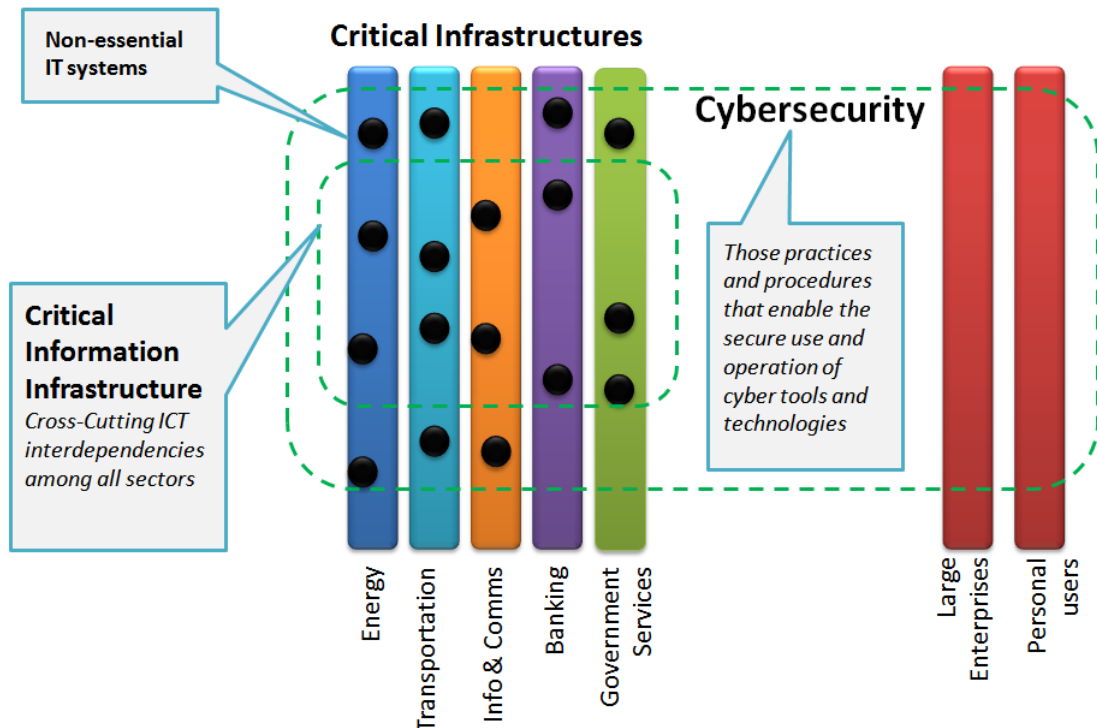


Figure 1.1: The role of Cybersecurity in non-essential vs. Critical Infrastructure. (Figure adapted from [2])

transportation, banking etc. count as CI domains, not all the software systems used in these areas (e.g., word processing tools used by the police department) can be categorized to be critical. According to [2], only infrastructure components, whose exploitation, or destruction, through natural disaster, technological failure, accidents or intentional attacks could have a debilitating effect on national security and economic well-being should be treated as CI. Although cyber security is important for “non-essential” software systems, CI components are more vulnerable to such attacks and therefore require special attention.

- *Supervisory Control and Data Acquisition:* Supervisory Control and Data Acquisition (SCADA), CCTV are components of CI. SCADA is a type of industrial control system and used widely in CI. SCADA monitors, controls, and gathers data from remote locations to control equipment and conditions. SCADA system includes both hardware and software. Hardware gathers data and feed into a SCADA software installed computer. Computer processes data recording and collecting logs of all events. Human Machine Interface presents data to a human operator to monitor and control process. Data acquisition is possible through sensors placed throughout the process or facilities. To relay data from the remotely located sensors, different hardware and software systems utilize various open communication protocols (e.g., OPC) to communicate data to Human Machine Interface.

- *Security in Critical Infrastructure:* CI is often built on SCADA and SCADA systems were never designed with security in mind. Consequently, data from SCADA systems is a prime target for cyber attacks due to the profound and catastrophic impacts they can have on our economy and on all aspects of our life [3]. This allows us to focus on the Cloud. Given the public awareness of CI and its importance, the public wants to be assured that these systems are built to function in a secure manner. Hence, a logical consequence is selecting the appropriate security means and developing such systems with proper documentation of the used security means.
- *Open Issues of Critical Infrastructure:* CI is built on very simple network device for example SCADA. SCADA was historically isolated from systems protected by security mechanisms and provides remote coordination of controls systems. SCADA systems were not designed considering being exposed to the Internet and were not considered security for open situation. This allows attackers to attack on the data from SCADA system. Thus, exposing to internet requires security against breaches that are possible in the Internet.
- *Critical Infrastructure in the Cloud:* To benefit from distributed nature of cloud computing and effectiveness in cost, service etc., CI is integrated in cloud computing. CI systems were accessed by only trained persons in the past but now it is integrated in cloud computing and exposed to public.
- *Open Issues of Critical Infrastructure-Cloud:* Previously, CIs were not openly accessed by users and were not designed for open access. Due to several benefits of cloud computing, CI is integrated in the Cloud. Integration of CI in the Cloud makes CI expose openly in the Internet. This makes CI-Cloud systems vulnerable and target of attackers when exposing to public. Thus, this open situation needs to be considered in security perspective. Security engineering of CI-Cloud applications needs to be considered.

1.2 Security Engineering

Security engineering is an engineering process that focuses on the security aspects while designing systems that need to deal robustly with possible disruption such as ranging natural disasters or malicious acts. It focuses on building systems to remain dependable in the face of malice, error, or mischance. As a discipline, it focuses on the tools, processes, and methods needed to design, implement, and test complete systems, and to adapt existing systems as their environment evolves. Security engineering requires cross-disciplinary expertise, ranging from cryptography and computer security through hardware tamper-resistance and formal methods to a knowledge of economics, applied psychology, organizations and the law. System engineering skills, from business process analysis through software engineering to evaluation and testing, are also important; but they are not sufficient, as they deal only with error and mischance rather than malice [4].

Many security systems have critical assurance requirements. Their failure may endanger human life and the environment (as with nuclear safety and control systems), do serious damage to major economic infrastructure (cash machines and other bank systems), endanger personal

privacy (medical record systems), undermine the viability of whole business sectors (pay-TV), and facilitate crime (burglar and car alarms). Even the perception that a system is more vulnerable than it really is (paying with a credit card over the Internet) can significantly hold up economic development. The conventional view is that while software engineering is about ensuring that certain things happen ('John can read this file'), security is about ensuring that they don't ('The Chinese government can't read this file'). Reality is much more complex. Security requirements differ greatly from one system to another. One typically needs some combination of user authentication, transaction integrity and accountability, fault-tolerance, message secrecy, and covertness. But many systems fail because their designers protect the wrong things, or protect the right things but in the wrong way. Getting protection right thus depends on several different types of process. So, we need to know what needs protecting, and how to do it [4].

Software Security Standards, Guidelines and Tools

Assurance is achievable applying appropriate security standards and tools in appropriate context. Applicability of security standards and tools in different context in the Cloud is a research area in cloud computing. Survey of existing processes and standards identifies security engineering activities, assurance activities, organizational and project management activities, risk identification and management activities [5]. These activities focus on secure software development process and have impact on security assurance.

Various software security standards are available to deal with security issues. Applying secure software development standards and tools contribute to data confidentiality, integrity, availability (CIA) and other security related issues. Based on these issues, we investigate the applicability of secure software development standards and tools (e.g. Security Development Lifecycle - SDL [6–8], Correctness by Construction methodology [9], Common Criteria Standard [10], ENSIA cloud security guidelines [11, 12], Computer Emergency Response Team - CERT [13–15], ISO/IEC 27001:2005- Information technology Security techniques - Information security management systems - Requirements [16, 17], ISO/IEC 27002:2005- Information Technology - Security Techniques - Code of practice for information security management [18] etc.) for cloud computing focusing on CI.

1.3 Research Open Issues

Requirements based security issues can be quite different between CI applications and (traditional IT) cloud applications but need to be considered in combination for the given context. For instance, a special challenge in developing (traditional IT) cloud application is that cloud systems are particularly vulnerable to security breaches. Typically characterized by the distributed nature of applications, and increased number of data transfer/storage issues. CI were not designed to be exposed to the Internet and often consist of limited resources of its (SCADA) components - these are significant difference of CI to traditional IT systems. Adoption of cloud technologies allows CI to benefit from dynamic resources allocation for managing unpredictable load peaks. Thus, integration of CI to the Cloud exposes to the Internet and has several open security issues that need to be addressed. For instance, data are not encrypted before being stored in the Cloud.

Existing secure software development standards, guidelines and tools are applicable in different context and helps us to address the open security issues. Choosing the appropriate software security standards, guidelines and tools will help us to overcome this problem. Thus, we will be investigating on

- Mapping of the security issues while considering CI to the Cloud and secure software development standards, guidelines and tools. We will define multidimensional taxonomy of the identified issues pointing to the secure software development means [19].
- Evaluating the applicability of the standards, guidelines and tools to CI [19].

1.4 Research Method

Our research approach can be categorized as a fact-finding empirical study, where we derive the facts based on our literature review. Security issues and requirements, that can be directly (empirically) observed are collected and abstracted to keep them applicable in general contexts. In the next phase, these facts are organized and categorized and in some cases complemented with other rationale ideas, in order to create a multi-dimensional taxonomy.

Motivating by selecting relevant standard, guideline and tool supporting CI in the Cloud, we will categorize standards, guidelines and tools into two dimensions (i) standards, and (ii) guidelines and tools. Based on the idea of Fletcher et.al. [20], we will sub-categorize these two dimensions to (i) security standards or guidelines and tools and (ii) software development standards or guidelines and tools. Some of the risks of our approach are that, the study could be biased on tenacity (issues are accepted because the idea has been accepted for a long time), intuition (acceptance without process of interpretation or assessment) and authority (accepted due to the high standing of the source).

We will use a show case to identify security issues and analyze the security standards which are useful in that particular context. By discussing how the standards and guidelines contribute to the security of the example application, we will qualitatively validate the results.

1.5 Contributions

Our contributions are two fold:

- a software developer knows which means is to be used for creating a secure CI cloud application that will help to mitigate specific security issues
- shed some light on the applicability of means for creating a secure software in a CI context

For this, our two scientific papers have been published and available in IEEE digital library. Firstly, a position paper with research issues, research ideas and goals, and secondly, a paper relating our use case to evidence gathering mechanism in multilayer cloud computing considering both legal and technical issues.

Venue : The 8th International Conference for Internet Technology and Secured Transactions (ICITST-2013), December 9-12, 2013, London, UK

- Sarita Paudel, Markus Tauber, Ivona Brandic. Security Standards Taxonomy for Cloud Applications in Critical Infrastructure IT. 2013 IEEE International Conference.
- Markus Florian, Sarita Paudel, Markus Tauber. Trustworthy Evidence Gathering Mechanism for Multilayer Cloud Compliance. 2013 IEEE International Conference.

Background

In this Chapter, we present some background on related domains and technologies to set up a ground for further discussion. For each research area described, we associate the research challenges related to security issues and their impacts.

2.1 Cloud Computing

Cloud computing is the use of computing resources (hardware and software) which are available in a remote location and accessible over a network. Cloud computing entrusts remote services with a user's data, software and computation. End users access cloud-based applications through a web browser or a light-weight desktop or mobile app while the business software and user's data are stored on servers at a remote location. Proponents claim that cloud computing allows companies to avoid upfront infrastructure costs, and focus on projects that differentiate their businesses instead of infrastructure. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand.

Cloud computing adopts concepts from Service-oriented Architecture (SOA) that can help the user to break users difficult business problems into services that can be integrated to provide a solution. Cloud computing provides all of its resources as services, and makes use of the well-established standards and best practices gained in the domain of SOA to allow global and easy access to cloud services in a standardized way.

Cloud Computing Characteristics

- *On-demand self-service*: Services such as email, network, server services and computing capabilities such as server time, network storage (as needed) can be provided without interaction with service provider [21]. Example of cloud service providers of on demand self service are: Amazon Web Services, Microsoft, Google, IBM.

- *Broad network access*: Cloud capabilities are available over network and accessed through standard mechanisms. These standard mechanisms promote use by heterogeneous thin or thick client platforms [21]. For example an organization team can access business management solutions using their smart phones, tablets, laptops, and office computers. Broad network access includes private clouds that operate within a company's firewall, public clouds, or a hybrid deployment.
- *Resource pooling*: Computing resources (e.g. storage, processing memory, network bandwidth, virtual machines and email services) are pooled to provide multiple consumers using multi-tenant model and combining different physical and virtual resources. These resources are assigned and reassigned dynamically according to customers' demand. Customers have no knowledge or control of resource location [21].
- *Rapid elasticity*: Cloud services can be rapidly and elastically provisioned. Services quickly scale out and rapidly released to the quickly scaled in services. Capabilities for provisioning are often unlimited and can be purchased in any quantity and any time [21].
- *Measured service*: Usage of computing resources can be measured, monitored, controlled and reported. Transparency of utilized services is provided for both cloud provider and consumer. Services use metering capability to control and optimize resource use. Electricity, municipality water, services are charged as pay-per-use. So, the more you utilize the higher the bill [21].
- *Multi Tenacity*: Multi Tenacity is advocated by the Cloud Security Alliance (CSA). Policy-driven enforcement, segmentation, isolation, governance, service levels, and chargeback/billing models for different consumer constituencies are needed. Consumers might utilize a public cloud provider's service offerings or actually be from the same organization, such as different business units rather than distinct organizational entities, but would still share infrastructure [21].

2.2 Cloud Computing Types

Large pools of resources can be connected to each other through private or public networks in cloud computing technology. It provides dynamically scalable infrastructure for cloud based applications, data storage etc. Depending on the accessibility of the Cloud, cloud computing is categorized in Private, Public, Hybrid and Community cloud computing [21, 22]. We will describe these types in this section.

Private Cloud Computing

Cloud infrastructure dedicated to an organization is a Private Cloud. It can be managed internally in an organization or by a third party, and can be hosted internally or externally. In many cases private Cloud is not shared with other organizations. Private Cloud are of two types [22]:

- *On-Premise Private Cloud*: Private Cloud hosted within an organization is called On-Premise Cloud. These Clouds are used for applications that require complete control and configurability of the infrastructure and security.
- *Externally Hosted Private Cloud*: Externally Hosted Private Cloud is hosted by a third party in Cloud Infrastructure. These types of cloud also used by one organization. Cloud environment is provided with full guarantee of privacy, and this environment is recommended for organizations that do not prefer usage of public infrastructure being aware of risks associated with sharing of physical resources.

Public Cloud Computing

Service providers host Cloud Infrastructure to make it available for general public. Examples of public cloud providers are Amazon AWS, Microsoft, Google. These providers own and operate Cloud Infrastructure and offer access over the Internet. Customers do not have visibility and control over Infrastructure. Customers share common Infrastructure pool with limited configuration, security protections and availability. These are more vulnerable than private clouds.

Cost for Infrastructure is spread over all customers sharing same Infrastructure so there is low-and pay-per-use cost. It also provides on demand scalability.

Community Cloud Computing

A Community Cloud is a multi-tenant cloud service shared among several organizations, and governed, managed and secured by all organizations or a third party. These Clouds are hybrid of private clouds, and targeted and operated for a specific group. Organizations can benefit from a public cloud with added privacy, security associated with private cloud. Community Clouds can also be on-premise or off-premise.

Hybrid Cloud Computing

Composition of two or more clouds (private, community or public) is Hybrid Cloud. Different models are combined together offering the advantages of multiple deployment models. Its architecture requires both on-premise resources and off-site server based cloud infrastructure.

2.3 Cloud Computing Service Models

Service delivery models in Cloud Computing are three types. They are Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS) [21, 23]. These service models are completed by an end user layer. The end user layer encapsulates end user perspective on cloud services.

Software as a Service

Software-as-a-Service (SaaS) is an end user application over a network on a pay-per-use basis. It is accessed through a web portal and service oriented architectures based on web service

technologies. The software requires no client installation, it requires just a browser and network connectivity. An example of SaaS is MicroSoft Office365 [24]. With Office365 you can get Word for a small monthly fee, with no client installation, the files are automatically backed up, software upgrades are automatically received and the software can be accessed from anywhere. If you do not need it just stop to pay monthly fee.

Platform as a Service

Software development companies or software developers use this service to run their software products. It provides platform to run an application such as physical servers, database software, Web servers. Building all of these ourselves is a time consuming and needs to be continuously monitored and updated. Platform-as-a-Service (PaaS) provides all of the platform software applications to execute them with no requirement for administration of the lower level components. All lower level elements of Infrastructure, Network Topology, Security and Load Balancers are done for us by the Cloud Service Providers. The Providers give us a fully functional OS with major platform software. E.g. Microsoft Windows Azure as PaaS can be used as a development, service hosting and service management environment. SQL Azure can provide data services, including a relational database, reporting and data synchronization. Both Windows Azure and SQL Azure are the key components of the Azure Cloud Platform. With this platform, we can focus on deploying our custom applications and can easily configure these applications to scale up or down as demands change.

Infrastructure as a Service

Infrastructure-as-a-Service (IaaS) covers individual servers, private networks, disk drives, various long term storage devices as well as email servers, domain name servers as well as messaging systems. All of these can be provisioned on demand and often include software license fees for operating systems and associated software installed on the servers. Organizations can build a complete computing infrastructure using IaaS on demand. For example, using Microsoft Windows Azure, we can set up new Windows Server and Linux virtual machines and adjust our usage as our requirements change. We only have to pay for the service that we use.

Dependency of these Cloud services is shown in the Figure 2.1. It shows different layers of models and connection to user.

2.4 Security

Security is an attribute of a system that reflects the ability of the system to protect itself from external attacks. These attacks may be accidental or deliberate. Computers are networked so external attacks are possible. For example:

- Installation of viruses and trojan horse,
- Unauthorized modification of a system or its data
- Unauthorized use of services

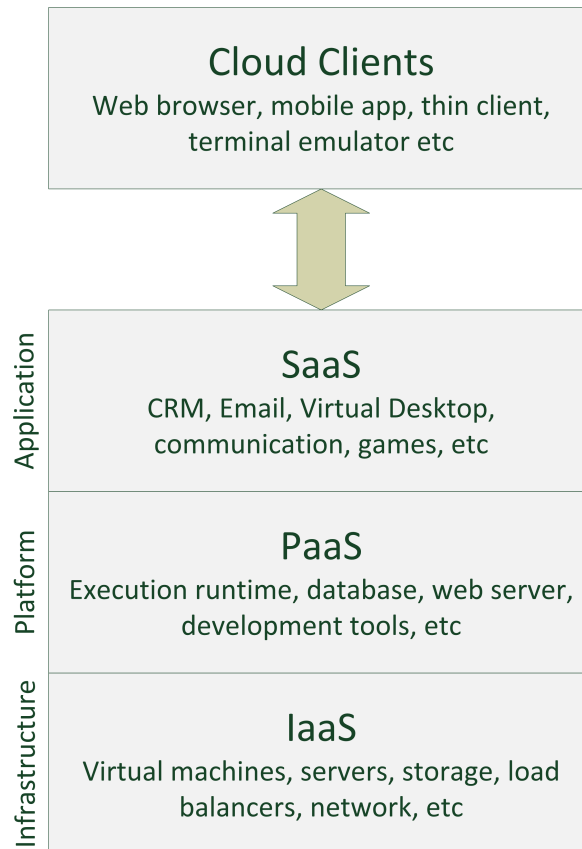


Figure 2.1: Cloud service models (Figure adapted from Wikipedia).

Military systems, systems for electronic commerce, and systems that involve the processing and interchange of confidential information must be designed to achieve a high level of security. For example, if an airline reservation system is unavailable then this causes inconvenience and some delays in issuing tickets. But if system is insecure and an attacker could delete all the bookings then it is impossible for normal airline operation to continue.

In any networked system there are threats to the confidentiality, integrity and availability of the system and its data. These threats are interrelated, e.g. if a system is unavailable and not able to update information that changes with time. Then this means that the integrity of the system may not be compromised. If an attacker succeeds and integrity of a system is compromised then the system is taken down to repair the problem. As a consequence, the availability of the system is reduced.

Security failure may lead to a loss of availability, damage to the system or its data, or the leakage of information to unauthorized people. Without a reasonable level of security, the availability, reliability, and safety of the system may be compromised if external attacks damage the system. If a system is unreliable, it is difficult to ensure system's safety and security, as they may be compromised by a system failure.

2.5 Design for security

It is difficult to add security to a system when it is already implemented. Therefore, designing systems for security is to be considered. For different applications different security design has to be taken into account for purpose, criticality and operational environment of the applications. For example, while designing military system, we need to adopt security classification model (secret, top secret etc.) and while designing personal data maintaining system, we need to take account of data protection law etc. [25].

Architectural Design

With inappropriate architecture of a system, it is hard to maintain confidentiality, integrity and availability of information in the system. While designing system architecture maintaining security we need to consider protection (how to protect critical assets against external attacks) and distribution (how to distribute assets to minimize effects of successful attacks). Security is to be considered in different layers of systems.

- *Platform-level protection*: It is a top level control access to platform. Platform normally support for maintaining integrity of files in a system, backups etc. This protection level involves: i) System Authentication ii) System Authorization and iii) File Integrity Management.
- *Application-level protection*: Application level protection comes next to platform level protection. It is a protection level built into an application itself. To access an application, a user should be authenticated and get authorization to view and modify data. It is protected as : i) Database login ii) Database Authorization iii) Transaction Management and iv) Database Recovery.
- *Record-level protection*: Record level protection is invoked after a user accesses an application and accesses a specific record. It checks whether the user is authorized to perform a requested operation for the record. It also involves record encryption, integrity checking. Thus, this protection level has three main protection categories: i) Record Access Authorization ii) Record Encryption and iii) Record Integrity Management.

Good practice

No hard and fast rules are available for system security. Different systems require different technical measures to achieve system security. There are general guidelines that are applicable in designing systems security. These guidelines encapsulate good design practice for secure systems engineering. Guidelines applicable to the software specification and design processes are discussed below.

- *Base security decisions on an explicit security policy*: A security policy sets out fundamental security conditions for an organization. It defines what security policy is to be provided, it does not define mechanisms to be used. For example, if you are designing

an access control system for hospital and hospital security policy states that only accredited clinical staffs have right to modify electronic patient records. In this case, we should check accreditation of a user to modify the system and reject modification from the people who are not accredited.

- *Avoid a single point of failure:* Try to avoid a single point of failure. That is, a single failure in a part of a system should not result in an overall systems failure. We should not rely on a single mechanism to ensure security, should employ several different techniques. For example, if we use passwords to authenticate users then there might be mechanisms to pre-register questions and answers with the system. So, after authentication users must answer questions before access. For integrity, keep log of changes to data.
- *Fail securely:* Safety-critical systems should always fail-safe and security critical systems should always fail-secure. For example, there is requirement that patient data should be downloaded to a system client before clinic session. This speeds up access and means that access to server. Normally, server deletes data at the end of session. If a server fails and data can be maintained by a client then a fail-secure approach is encrypting all patient data stored in the client so unauthorized user cannot read the data.
- *Log user actions:* Maintain a log of user actions recording who did what, the assets used, time and date of actions. We can analyze logs and detect potentially anomalous actions. Tools can scan log and find anomalous actions, and thus help us to detect attacks and trace how the attackers gained access to the system.
- *Use redundancy and diversity to reduce risk:* Redundancy means if we maintain more than one version of a software in a system. Diversity means, different versions of a software should not rely on same platform and implemented using same technologies. A platform or a technology vulnerability will not affect all versions and lead to common failure.
- *Validate all inputs:* Attack in a system can be caused by providing the system with unexpected input, which cause system to behave in an anticipated way. This may cause system crash, resulting in a loss of service and input could be malicious code executed by system. Buffer overflow, SQL poisoning are common attacks.
- *Design for deployment:* Deployment of a system means configuring software to operate in an operational environment, installing the system on the computers in that environment and then configuring the installed system for these computers. When a system is configured incorrectly while deploying in operational environment many security problems can be raised. Therefore, systems should be designed considering facilities that simplify deployment in the customer's environment and check for potential configuration errors and omissions in the deployment system. There are different ways for deployment support such as: i) Include support for viewing and analyzing configurations, ii) Minimize default privileges, iii) Localize configuration settings, iv) Provide easy ways to fix security vulnerabilities – update system to repair security vulnerabilities that have been discovered.

- *Design for recoverability*: Although a system is maintaining security, it should be designed with the assumption that a security failure could occur. Therefore, we should think how to recover from possible failures and restore system to a secure operational state.

2.6 Open Web Application Security Project

Open Web Application Security Project (OWASP) [26] is worldwide not-for-profit organizations focused on improving the security of software. It is dedicated to conceive, develop, acquire, operate, and maintain trusted applications. All of the OWASP tools, documents, forums, and chapters help in improving application security. The OWASP community includes corporations, educational organizations, and individuals and create freely-available articles, methodologies, documentation, tools, and technologies. It is also an emerging standards body for example, OWASP Application Security Verification Standard (ASVS) Project is to normalize the range of coverage and level of rigor available in the market when it comes to performing application-level security verification. Goal is to create a set of commercially workable open standards that are tailored to specific web-based technologies. OWASP lists top attacks, vulnerabilities, countermeasures etc. and helps to avoid possible attacks.

Related Work

To put our work in the context of the state-of-the-art, we are discussing the existing work on exploring security issues of the Cloud and CI, discussing security requirements and assessing, developing a taxonomy to select appropriate means for secure software development process and implementing standards to build secure software. We do a study on some commercial products that helps to develop software and the security objectives, types of security threats in cloud computing. We also investigate on CI specific issues in Chapter “Show Case and Security Issues Analysis”. Most of the existing approaches do not address issues related to CI and those which do, do not provide any guidelines to select appropriate software security standards, guidelines and tools. We are extending these existing approaches to select appropriate means for secure software development focusing on CI IT in the Cloud.

3.1 Security Requirements and Assessment

Youchan Zhu et.al. [27] analyze security research status in the Cloud and provide security solution. The authors report that the abuse of cloud computing resources and environment security are the common security problems in the Cloud. Our approach also reports some open security issues but focuses in CI and the applicability of security standards and tools. Kui Ren et.al. [28] outline various critical security challenges such as data service outsourcing security, computation outsourcing security, access control, trustworthy service metering, and motivate for further investigation of security solutions by pointing out their importance. Our approach is also to motivate for further investigation of security standards, guidelines and tools and furthermore their applicability in the context of CI.

In [29] ENISA generalises security issues of cloud computing from Critical Information Infrastructure Protection (CIIP) perspective. Additionally, authors discuss on the risk assessment and security measures. This work is also similar to ENSIA only on outlining open issues of CI in the Cloud. But our main contribution is selecting security standards and tools based on the taxonomy and evaluating the effect of the selected software security standards and tools in

the show case. Abbadi et.al. [30] discuss security challenges and security requirements moving from cloud untrusted infrastructure to trustworthy cloud CI. They discuss challenges and requirements focusing on cloud resource management for virtual infrastructure, security threats from cloud insiders, cloud user-centric security model and cloud infrastructure self-managed services. This work focuses not only on issues in CI but also investigate on the applicability of some popular security standards and tools in the CI. Similarly, Younis et.al. [31] explore security issues of secure cloud computing for CI Providers and investigate security requirements for the CI Providers. Our research approach similarly outlines the open security issues in the CI but additionally, we investigate the applicability of software security standards and tools for CI IT in the Cloud.

Management of cloud services is complex and error-prone. Assurance of error-free actions is difficult. Accidental or intentional actions of cloud administrators could cause a serious threat to integrity and confidentiality of data. Trusted computing helps to mitigate threat. Cloud exposes internal details of the Cloud infrastructure, hinders fault tolerance and load balancing flexibility and performs poorly. Trusted Platform Model (TPM) is a secure coprocessor deployed in every nodes in the Cloud and stores a strong identity and a fingerprint of the booted software in each node. It provides building blocks for constructing trusted services by securing data confidentiality and integrity against insiders. TPM also has some limitations (e.g. scalability bottlenecks to cloud services, over-expose to cloud infrastructure). Existing technology relies on TPM and due to its limitations, the technology is not perfectly suited to the Cloud service requirements. Santos et.al. [32] present Exacalibur, a system that provides new trusted computing abstraction called policy-sealed data addressing the limitations. Policy-sealed-data allows data to be sealed and unsealed only by the certain nodes. It allows encryption according to the encryption policy and allows decryption only by the nodes whose configuration match the policy. Furthermore, authors discuss on the design goals and assumptions to overcome limitations, cryptographic policies, trusting monitor, scalability and fault tolerance of monitor etc in Exacalibur. This related work discuss trusted computing providing policy-sealed-data in common cloud systems. Our focus is on trusted computing in the context of CI in the Cloud.

Many security considerations are taken into account in Service Level Agreement (SLA) to ensure secure software development. SLAs state terms of use and penalties in case of violations. SLAs are assumed to be guaranteed at the application layer. Assurance in SLA convince the cloud customer on security issues. Balachandra Reddy Kandukuri et.al. [33] address SLA security risks such as privileged user access, regular compliance, recovery, long-term availability and security at different level such as server access security, client access security, database access security, data privacy security. Authors also conclude that security policies, methods and their implementation has to be taken into consideration in SLA. It requires motivation of research on security standards. We are not directly working on SLA but our work is also on the investigation on evaluation of software security standards and tools to support security issues in the cloud in CI context.

Self protective system are self-adaptive. Self-adaptation behavior of a system is due to their internal configuration parameters in response to a changing environment. Self-adaptation can be achieved by autonomic management of facets of a systems constituent components. Autonomic Computing Systems are self-adaptive systems that are capable of modifying their behav-

ior based on changes in its operating environment. Thus, autonomic system maintain required Service Level Agreements (SLA), protect the execution of the system from external attacks or prevent and recover from failures. These autonomic systems are developed as control loops that monitor and analyze system's execution and then plan and adopt the system's environment if needed while executing changes. Thus, automated and integrated intelligent strategies for provisioning of resources to offer secure, reliable, and cost-efficient services is a challenging task. Buyya et.al. [34] identify open issues of autonomic autonomic resource provisioning and also present management techniques for supporting SaaS applications hosted on Clouds. There is no existing work addressing these open issues while developing secure software in the cloud in CI context. We want to bridge this gap by addressing the issues in the development of secure cloud applications considering CI context.

Current implementation of cloud services do not meet high assurance. High assurance of cloud computing presents precise requirements and many challenges to normal computing. There are various challenges in bringing the cloud and high assurance together. Chandrasekaran et.al. [35] explore challenges of high assurance cloud and discuss how they may be able to overcome. Authors categorize these challenges in four groups: i) virtualization and the loss of attribution that accompanies a highly virtualized environment, ii) loss of ability to perform end-to-end communications, iii) need of appropriate encryption and comprehensive key management process for public key infrastructure, as well as session and cryptographic keys, and iv) monitoring and logging for attribution, compliance and data forensics. Authors also discuss a high assurance architecture and its requirements which challenges to cloud environment. From this related work, we know the open issues of high assurance cloud computing but there is no existing approaches for developing high assurance cloud applications addressing these open issues. To bridge this gap, we will not only address these issues in design phase but also implement security standards and best practices to cover the precise requirement of cloud applications in CI context in our taxonomy.

3.2 Classification of Secure Development Means

Cloud carrier represents a network which provides connectivity and transport of services between cloud providers and cloud customers. Lenkala et.al. [36] present a risk assessment framework to assess security of the Cloud carrier by analyzing the OS vulnerabilities of the routers in network. The framework characterizes security metrics including Confidentiality, Integrity and Availability (CIA), and compare the security of cloud carrier connecting to multiple cloud providers. This related work is motivated by the fact that security of applications deployed on cloud infrastructure depend on secure cloud carrier. Goal of our research is also to investigate on secure software development in CI focusing on CIA and developing a taxonomy that maps different security standards and tools to security issues in the cloud depending on their applicability.

Information security has to be addressed in design, development and deployment of software applications to ensure high level of trust. Fletcher et.al. [20] provides guidelines and practices for secure software. Authors discuss various guidelines and practices grouping in two parts i) security standards and best practices (e.g. ISO/IEC 27002, ISO/IEC 27001), and ii) software

development standards and best practices (e.g. Security Development Lifecycle-SDL, Common Criteria-CC,), and propose a set of guidelines for secure software development (e.g. managing the software development process, security functions to be built into-applications). This work is also similar to it in the sense, the output of this research will also provide standards, guidelines and tools but for addressing specific issues in CI. We are motivated by this existing work and also categories i) standards to security and software development standards, and ii) guidelines and tools to security and software development guidelines and tools. We however consider this as subcategories for a) standards and b) guidelines (see later). Reason for this is that in CI specific areas explicitly require standardised approaches where others don't.

Taxonomy is the science of classification and categorisation of things based on the predefined system. It is presented as hierarchical tree-like structure. Complex applications compose and execute complex workflows. Yu et.al. [37] propose a methodology or a tool for characterization and classification of workflow management approaches to build and execute workflows of complex applications computing. Authors also present a survey of existing workflow systems to mapping the methodology. We can get an idea of classification and mapping from this approach. Our work will be in the applicability of security standards and tools in the context of CI. It needs categorization and classification of existing security standards and tools based on the taxonomy developed from requirements. Similarly, Dukaric et.al. [38] propose a unified taxonomy and an Infrastructure as a Service (IaaS) architectural framework. Furthermore, authors use the proposed taxonomy and framework for evaluating different IaaS architectures. The unified taxonomy comprises seven essential layers (core service layers, support layer, value-added services, control layer, management layer, security layer and resource abstraction) and each of them consists of several components. The taxonomy is used to identify and classify the fundamental IaaS into layers or categories, and also used to design the IaaS framework. We can get an idea from this existing work to develop and map taxonomy. But we will develop a taxonomy based on the identified security issues and security requirements in CI to point out different existing security standards and tools supporting CI and use the taxonomy for mapping existing standards and tools based on evaluation.

Similarly, Savola [39] conduct a survey of security metrics approaches and propose a security metrics taxonomy for Information and Communication Technology (ICT) product industry based on the survey. Security metrics of the proposed taxonomy incorporates i) organizational information security management and ii) product development. This work is also similar in the sense, while defining taxonomy we will consider security metrics for requirements but additionally we will have security issues in CI.

3.3 Implementing software development standards

Management of security requirements is a complex task. Requirement engineering is a critical and hard task to develop a secure software. CC provides assurance of specification, implementation and evaluation process of IT security product. In [40] Razzazi et.al. describe design and implementation of Evaluation Process Management Software for security evaluation. To evaluate IT products using CC, authors describes the roles of developers, evaluators and administrators and their activities in evaluation. Additionally, they also present flow of evaluation process. This

existing work is a good example for this work because we are also going to implement security standard but the standard will be selected after doing survey and CI relevant.

An approach for addressing security in software design, development, and testing process has been reported in [41]. Here the object of the study was a operating system (Caernarvon). The authors share the experience of using Common Criteria Evaluation Process targeting Evaluation Assurance Level 7 (EAL7). Our investigation is also similar in the sense we will address security issues by implementing security standard but in the context of CI. And the relevant security standards will be selected after doing survey.

A large percentage of security problems such as SQL Injection, Cross-site Scripting, Buffer overflow are caused by string-based code injection vulnerabilities. Most of these vulnerabilities are due to implicit code creation through string serialization. Knowing the cause of vulnerabilities and applying countermeasures are needed to be considered in implementation of programming languages. Johns et.al. [42] analyze vulnerabilities underlying mechanisms and propose an approach for secure code generation providing strict separation between data and code. This related work is focusing on general web application, but our focus will be in the cloud applications and we will be developing a taxonomy for secure code generation by implementing security standards in CI context.

Additionally to the academic papers cited above recently started research project are dealing with related issues¹.

3.4 Commercial Products

Checkmarx [43] develops Static Code Analysis solutions to introduce security into Software Development Lifecycle (SDLC). The automated scanning technology provided by Checkmarx enables developers to easily scan uncompiled code in major programming languages and helps to systematically remove risk. Thus, this tool supports security in software development but is not addressing for cloud applications. For this gap we are developing a taxonomy to choose appropriate means to introduce security in cloud application and additionally in CI context. Similarly, there is another tool, [44] Crunchbase provide a secure code generator that generates cryptographically secure code but not for high assurance cloud applications. To bridge this gap, our work will develop a taxonomy to generate secure code to produce secure CI cloud applications. Besides all the academic approaches, available commercial tools are also not providing means for producing secure cloud applications in CI context.

3.5 Security Objectives in the Cloud

Security objectives define what we are trying to achieve from a security perspective, and these are also called security requirement. In cloud platform, attack vectors and security issues are mapped to one of the security objectives in the Cloud. We also do study of security objectives

¹For more information see : www.a4cloud.eu, cumulus/project.eu. These projects are also focusing on CI with goal to increase trust in the Cloud

in the Cloud. Six security objectives of cloud platform presented by Saripalli et.al. [45] are the following:

- *Confidentiality*: Defines a set of rules to limit access to information. It prevents access of sensitive information by wrong people. It only allows to access only by right person. For example credentials for accessing bank account.
- *Confidentiality*: Integrity is the assurance of trustworthy and accurate information. It maintains consistency, accuracy and trustworthiness of data over life cycle. Must be ensured that data cannot be changed by unauthorized person and not changed in transit.
- *Availability*: It defines the availability of information or services for authorized user. It can be provided by ensuring maintenance of all required hardware and providing sufficient communication bandwidth.
- *Multiparty trust*: Multi-tenant infrastructure [46] refers to unrelated users of shared computing infrastructure which is a fundamental characteristic of cloud computing. The key security between multi-tenant infrastructures is to establish trust with providers, users, trust the exchanging data. Shared Infrastructure, multi-provider infrastructures are trusted by customers. It includes trusted cloud or share infrastructure defining end-to-end reference models for deployment.
- *Mutual Auditability*: Auditability [47] is the degree of tracing and auditing transactions through a system. As an administration verify that users perform actions themselves and is not completed by administrator or anybody else. So users can verify their actions performed by them. An auditor can determine who performs and complete which action.
- *Usability*: When we talk about usability in the Cloud, one is about the usability of cloud applications and the other about cloud computing environment. People using online applications are using cloud applications. For example: Apple iPhone has thousands of cloud applications (e.g. An application that takes users dictation and provides speech-to-text recognition). Therefore, cloud computing is everywhere the Internet is. It benefits the Cloud usability. Thus, usability of cloud application is increasing with the increasing order of online applications users. Cloud Computing Environment provides a single access point for all the computing needs of users, including access to computing and infrastructure resources, processes, applications or services, and supporting network infrastructure. Thus usability of the Cloud computing environment is complicated. It depends on application architecture, design, characteristics of applications like security requirements, availability requirements, amount of data transfer etc. and many more factors.

3.6 Security Threats in the Cloud

While moving applications from dedicated hosting infrastructure to shared infrastructure leased from the Cloud, security remains a sticking point [48]. Cloud hosting providers have less control over the construction, operation and auditing of the leased infrastructure than they own. As a

result, cloud-hosted infrastructure is less secure than the self-hosted infrastructure. Transitioning to cloud-hosted infrastructure must be secure. Threat events, attack surface, access point of each threat are identified in threat modeling and it helps analyzing associated risks and developing mitigating strategies [45]. These threats are of different types. Molnar et.al. [48] expose new cloud security threats related to:

- *Technical*: These threats are the technological threats. Misconfiguration of technology, vulnerabilities in technology and other technical reason cause technical threats. Technical threats are the cyber-attacks to software and hardware. Catalogue of technical threats are available from different research groups. Assessment for the catalogue of threats need to be considered.
- *Contractual*: Threats caused by cost-overflow, deceptive billing (billing tenant for more resources than actually consumed or less efficiently increase consumption of billable resources), switching cost and bankruptcy (CSP may go out of business, users will lose access to applications and data and unable to move before cloud provider's infrastructure goes offline) caused by attacks.
- *Jurisdictional*: Indirect legal coercion e.g. cease and desist requests sent to the Cloud infrastructure provider; direct and indirect jurisdictional exposure, exposure to "secret searches"(sealed search allow law enforcement to search the tenant's systems).
Virtual service/data migration outside of accepted jurisdictional boundaries, e.g., critical infrastructure or eGovernment data leaving the EU.
- *Organizational*: These threats are related to human resources. If human resources are not suitably audited, screened or trained then there could be some organizational threats.

The security issues arising from the security threats from Technical, Contractual, Jurisdictional and Organizational are needed to be addressed.

Research Methodology

In order to be able to evaluate the software development standards for Critical Infrastructure IT in the Cloud, the evaluation study has been carefully planned with the following research question: *What kind of support can be provided to software developers and security engineers trying to adopt existing security standards, guidelines and tools?* We have followed a descriptive study which attempts to describe systematically the set of available standards and their applicability when dealing with security issues. This means, we base our results on qualitative analysis of security standards and guidelines, thereby considering their applicability in the context of CI.

Research Agenda

The primary research goal of the proposed thesis is to investigate on secure software development in Critical Infrastructure in the Cloud focuses on data confidentiality, integrity, availability (CIA) and other security related issues. This is particularly important as concerns about security has made it difficult for many large companies to trust outside services to handle critical data and computing tasks. A further goal is to investigate the applicability of secure software development standards, guidelines and tools for cloud computing. Overall research issues addressed by our work are:

- Evaluation of available software security standards, guidelines and tools to support CI and the Cloud.
- Development of a multidimensional taxonomy based on open security issues for CI in the Cloud to point out multiple standards, guidelines and tools, and
- Mapping of security requirements to the available standards, guidelines and tools based on evaluation.

Corresponding results will eventually also be useful for service assurance when documenting the used software development security standards, guidelines or tools.

4.1 Phases of Research

Our research approach can be categorized as a fact-finding empirical study, where we derive the facts based on our literature review. Security issues and requirements, that can be directly (empirically) observed are collected and abstracted to keep them applicable in general contexts. These facts are organized and categorized, and in some cases complemented with other rationale ideas, in order to create a multi-dimensional taxonomy. Figure 4.1 is the graphical representation of different phases of Research.

Phase 1 *Problem Analysis*: In the first phase, we conduct a literature survey and implement an example CI-Cloud application. The goal is to get an overview of available standards, guidelines and tools, and identifying a list of Cloud security issues from literature survey. The implementation helps in understanding the security issues in an practical example. It is described in detail in Chapter 5.

Phase 2 *Generalisation and Interpretation*: This phase of research is devoted to analysis of the collected data on available standards, guidelines and tools. We study most popular secure software development standards, guidelines and tools. We also classify and categorize them. We map the identified list of the security issues in phase 1 to the standards, guidelines and tools depending on the applicability of the standards, guidelines and tools. It is presented in a multi-dimensional taxonomy and described in detail in Chapter 6 and 7.

Phase 3 *Validation of results*: In order to test the multi-dimensional taxonomy of the results in phase 2, we select an available standard from the multi-dimensional taxonomy and evaluate the mapping of cloud security issues and the selected standard. Thus, in this phase we are testing how well is the taxonomy and the standard for secure system development and security engineering that helps to make a secure system. Details can be found in Chapter 8.

4.2 Literature Survey

In the first phase of our research, we conduct a literature survey. The primary goal is to gain an understanding on the fundamentals and state-of-the art of security issues for Critical Infrastructures in the Cloud. At the same time, the study also helps in learning the definitions of the concepts and gives access to latest approaches, methods and theories on cloud security.

The challenge is to ensure that the used sources of information are reliable. Knowing what are the assumptions, restrictions and open questions that make possible to evaluate the reliability and applicability of the information [49]. Prerequisite for reliability and sharing information is that we use materials in peer reviewed journals and conferences and information from first hand sources (e.g., the owner of the standards like Microsoft). Additionally, journals, books and standards defined by official bodies are excellent sources.

The result of literature survey is a collection of currently available security standards, guidelines and tools for secure software development. The goal of literature survey is to identify a list of security issues in the cloud in CI context. We relate the identified issues to our show case

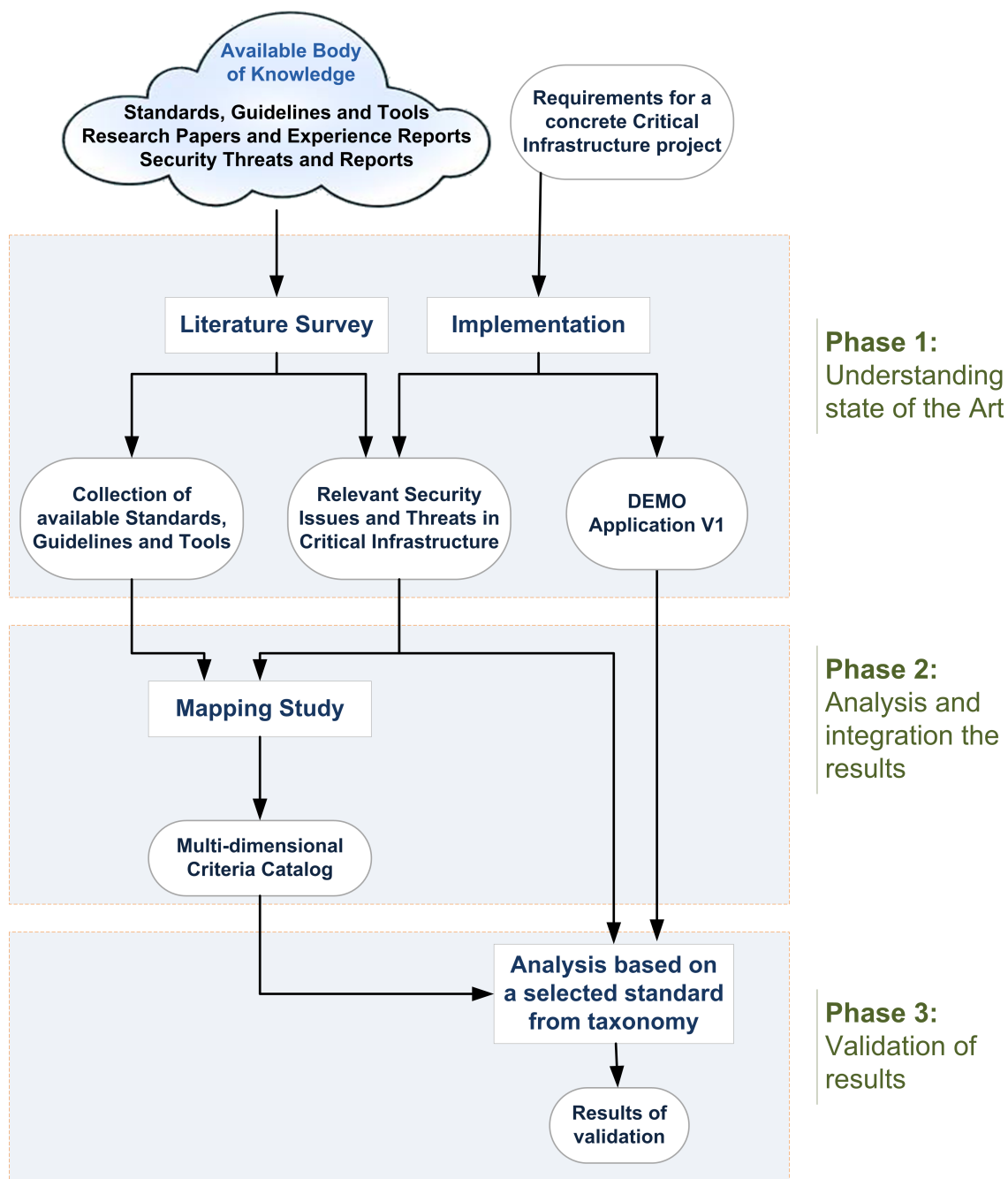


Figure 4.1: Overview of research approach– depicting the three phases and the activities in the different phases.

but there are some issues which are not related to our show case but related to CI context in the Cloud. Details can be found in Chapter 5.

4.3 Show Case Implementation

To gain first hand experience and understand the challenges from the developers perspective, we developed a show case application. The process of development confronts the developer directly to security issues which arise throughout the systems life-cycle: from concept and inception, through design, implementation, testing, and deployment, to maintenance and re-engineering.

Capability to design systems which will meet security goals comes through the experience of building such software. Our show case explores how suitable levels of assurance can be achieved through combining architectural detail, and application security measures. Central to these considerations is concerned which requirements are met with well-established tools, which risks can be addressed through novel technologies, and which must be mitigated by other means.

The example application is a show case - a client/server application with a web front-end. The goal of implementing this piece of software is to get first hand experience of building CI applications and to better understand the security issues. Details about the show case can be found in Chapter 5.

4.4 Mapping Study

After identifying a list of Cloud security issues, we also do a study on popular software security standards, guidelines and tools. We map the security issues to the different standards, guidelines and tools. This mapping depends on the applicability of the standards, guidelines or tools to mitigate the individual security issue. We present this mapping in the form of a multi-dimensional taxonomy. We classify these means of secure software development into standards, and guidelines and tools and sub-categorize them into software development and security.

The result of mapping study is a multidimensional taxonomy pointing out the appropriate security standards, guidelines and tools. This is the summary of Chapters 6 and 7, where the results are presented as a matrix. Detail of categorization and classification of secure software development means can be found in Chapter 6 and mapping of these means and the security issues can be found in Chapter 7.

4.5 Analysis based on a Selected Standard

After having defined the multi-dimensional taxonomy of the standards, guidelines and tools, we select a standard from the taxonomy, evaluate the taxonomy based on the standard. We select SDL Threat Modeling Tool and apply it in our show case. We use the data flow diagram of the show case (demo application) for evaluation.

At first, we convert security issues of the show case to Cloud security issues, select some of the issues and evaluate how SDL Threat Modeling Tool helps to mitigate the selected security issues. Detail of this evaluation can be found in Chapter 8.

4.6 Threats to Validity

The results of our research provide a first step towards a more systematic and extensive study, where the validity of the results can be generalized. The extent to which the results of our study can be generalized to other applications and other security issues is restricted. It is therefore not straightforward and not directly possible to transfer the results to other situations with similar issues, requirements and security features.

Most of the observations and conclusions are personal judgments of the author and these may have unconsciously changed the criteria used to make judgments. This can also be an issue with self-report measures given at different times. This kind of threat to validity is called „instrumentality“ [50] and it is highly relevant for our study.

Show Case and Security Issues Analysis

In this Chapter, we first develop a show case for cloud applications in the context of Critical Infrastructures. We then identify cloud security issues through a literature review and a show case analysis, relate these security issues to the show case and define Anticipated Result for each of the issues.

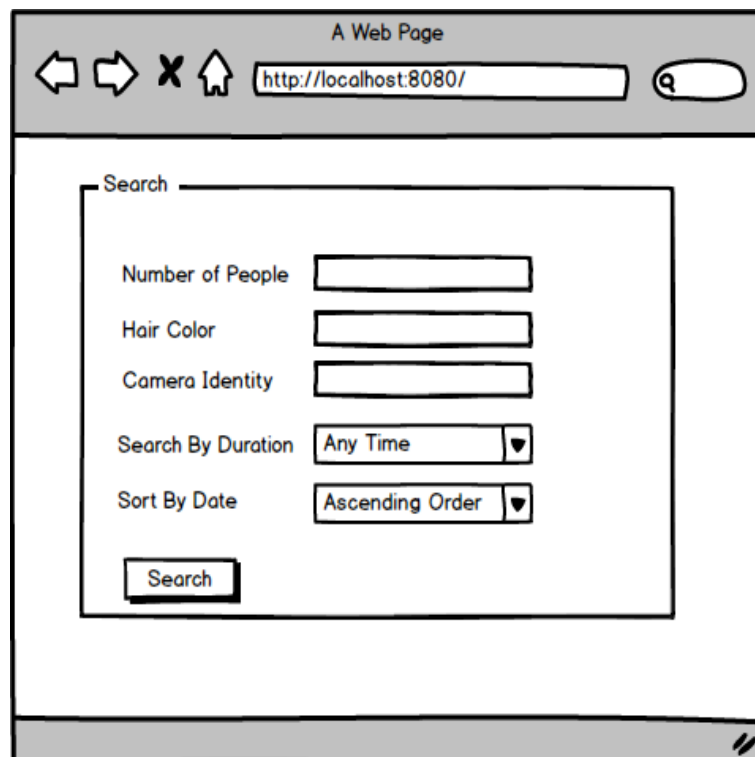
5.1 Show Case description

Motivation

Our motivation of developing show case is developing an application for Critical Infrastructure (CI) and the Cloud, and getting familiar with the security issues for CI and the Cloud. Our show case is a Client/Server application. It is developed for the use of public safety. For example, information retrieved from the videos of CCTV cameras are used for public safety. It is used for retrieving images and corresponding metadata from streaming videos, saving in database and searching images by metadata. By keeping usability or flexibility in mind we develop a web application connected to the elastic database in the Cloud for search facility. In this case, we could reuse this module in other application as well, i.e. it is an independent module. We deploy Web application and Server in real Cloud environment (VMWare : <https://cellmgr.qcloudwise.com/cloud/>) and we suppose Client run in surveillance camera. So, we collect information about CCTV cameras from different companies like Axis webcam, Bosch webcam, Sony webcam and eneo webcam, and find out the computation power of CCTV cameras in average. These results show that CCTV cameras have computational power. In general, CIs have some potential issues such as limited computational resources for example, insufficient or limited computational power [51]. Cloud is computationally powerful, so we have to balance these two things while establishing connection between CIs and Cloud [52]. Different types of storage systems in the Cloud have own unique capabilities and way of communicating [30]. So,

we have to consider the type of storage system and establish communication. Thus, we consider such issues while developing the showcase. Three modules of this application will be discussed in next section.

Mockups of our show case are in Figures 5.1, 5.2 and 5.3. Figure 5.1 is the mockup of search form. We can search the images by number of people in the images, hair color of the people, camera identity that has captured that particular video from which the images are extracted, search by duration for example: Any Time, Last 24 hours, Last week or Select Date Range. The result from the search can be displayed latest first or oldest first by choosing Sort by Date option.



The image shows a mockup of a web browser window. The title bar reads "A Web Page". The address bar contains "http://localhost:8080/". The main content area features a search form with the following fields and options:

- Number of People:
- Hair Color:
- Camera Identity:
- Search By Duration: (dropdown arrow)
- Sort By Date: (dropdown arrow)
- Search:

Figure 5.1: Search form for all available options

Figure 5.2 shows select date range option in search by duration field in the search form. When the select date range option is selected then start date and end date fields are available in the search form. User can choose start date and end date from calender.

Figure 5.3 is mockup for displaying search result. Boxes in left side are the images and text on right side is description of the images. Description of the images contains date on which image is captured, camera identity, number of people in images and hair color of the people in the images. This detail description is repeated for individual image.

A Web Page

http://localhost:8080/

Search

Number of People

Hair Color

Camera Identity

Search By Duration

Start Date

End Date

Sort By Date

Figure 5.2: Search form when select date range option is selected in Search By Duration

A Web Page

http://localhost:8080/searchResult

Welcome to Surveillance Search!

Picture displayed in left side and Metadata listed below are extracted from streaming video.
Date: 10/1/13 5:56 PM
Camera: 200
People: 5
Color: black

Picture displayed in left side and Metadata listed below are extracted from streaming video.
Date: 10/1/13 5:56 PM
Camera: 200
People: 5
Color: black

Picture displayed in left side and Metadata listed below are extracted from streaming video.
Date: 10/1/13 5:56 PM
Camera: 200
People: 5
Color: black

Figure 5.3: Search result

Modules

- *Client:* Netty Client is a data entry client. Data entry client extracts images and corresponding metadata from streaming video and sends them to Server in the Cloud. We are designing client in such a way that the client is connected to several cameras and gets streaming videos from the cameras. Client extracts images and corresponding metadata from these videos and sends to the Server in the Cloud. For each image and its corresponding metadata client sends a request to the Server. Thus, number of requests sent by client depends on the number of images retrieved from streaming videos. For each of the requests client receives response from server after successfully saving in elastic database.
- *Server:* Server receives requests from Clients and gets images and corresponding metadata for each request. Server is connected to elastic database where the images and metadata are to be stored. Server saves the received images and corresponding metadata from client to the elastic database. Server sends response to the client after saving data successfully in the elastic database.
- *Web Application:* Web application is deployed using apache tomcat and connected to the server. It is used to search the images saved in the database. User can browse search form on web and use search feature. User enters metadata of the images in the search form and submits the search form. User can search images by hair color, number of people in image, camera identity and duration. Duration feature is used to select dates. User has four options to select some start and end date to search records in database. Four options are Any Time, Last 24 hours, Last week and Select Date Range. If a user chooses Any Time, it shows all saved records. Last week option is to get data of last seven days. If the user chooses Select Data Range option then the user can choose start and end date from calendar. When a user provides available metadata and submits search form then system searches the images based on the metadata and displays result on the browser.

Figure 5.4 is a screenshot of the web application. Search form is depicted on left and result of the search is displayed on right. Result contains images and description of the individual images. Description contains date, camera identity, number of people, hair color of the people in the image. Three images and corresponding description is displayed per page. There are next and previous buttons on the bottom to browse more images from the result.

User Groups

- *Client for data collection:* An automated client connected to the CI, collects the data on a regular basis from CCTVs to retrieve images of streaming videos. These data items are sent to the Server and saved in elastic database.
- *Client for browsing data:* A browser based application is used by users to search and navigate through the data in the database. This application allows to send queries to the server based on the metadata of the saved images.

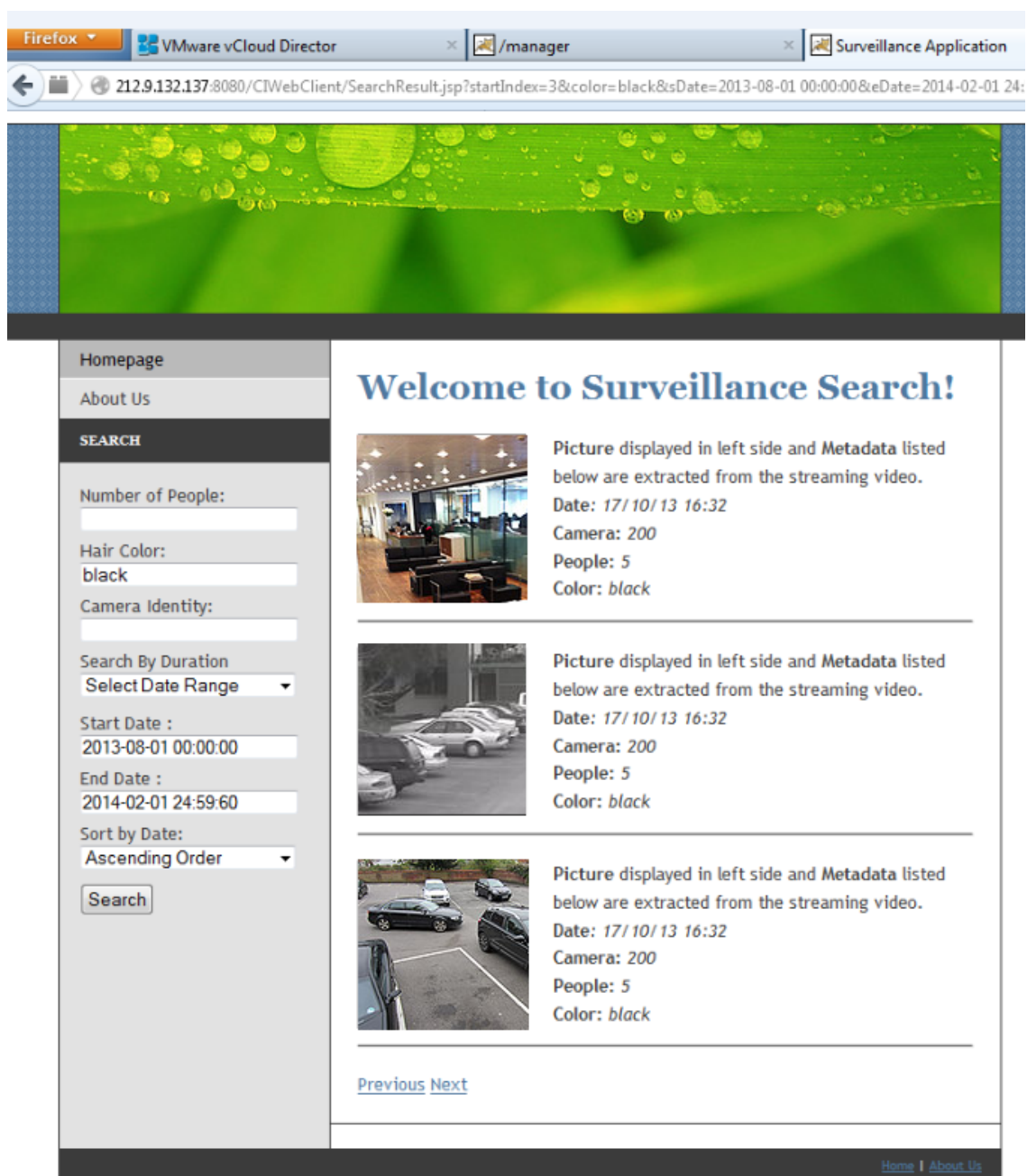


Figure 5.4: A screenshot of the browser application to search for surveillance images.

5.2 Technologies and Architecture

Technologies

We are using programming language java and Netty java framework to develop this application. Xuggler java library is used to extract images and metadata from streaming videos. Apache tomcat is used to deploy the web application. List of technologies we have used are the following:

- Java Programming Language 1.7
- Web server: Tomcat 7
- Desktop Client/Server: Netty framework 4.0
- Images and metadata extractor: Xuggler 5.4

Architectural Style

Netty Server is connected to Netty Client and elastic database in the Cloud. Web application to search images is also connected with elastic database. Direct connection of web application decreases load in Netty Server. Netty Server is connected to database only to save images and corresponding metadata in the database. Architecture of this application is shown in Figure 5.5.

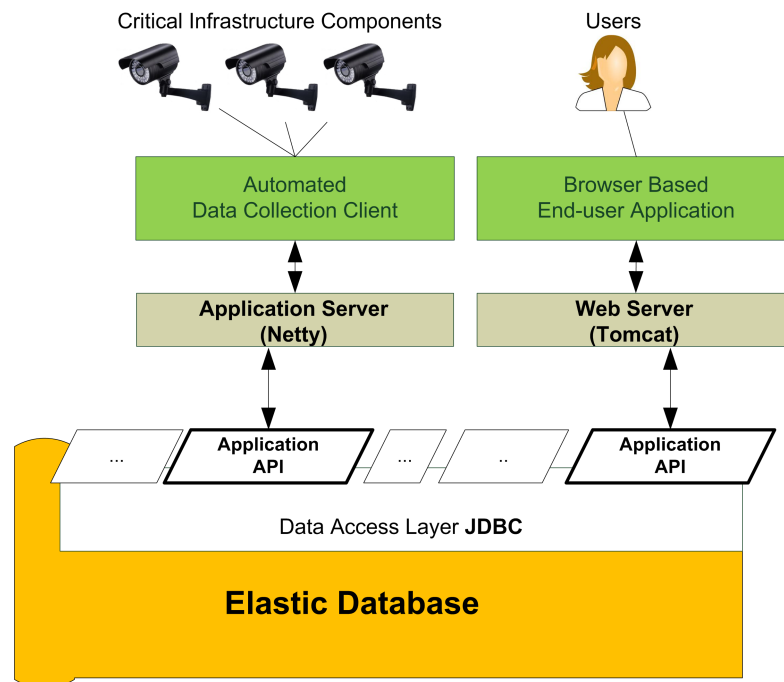


Figure 5.5: Architecture of the client-server and web application.

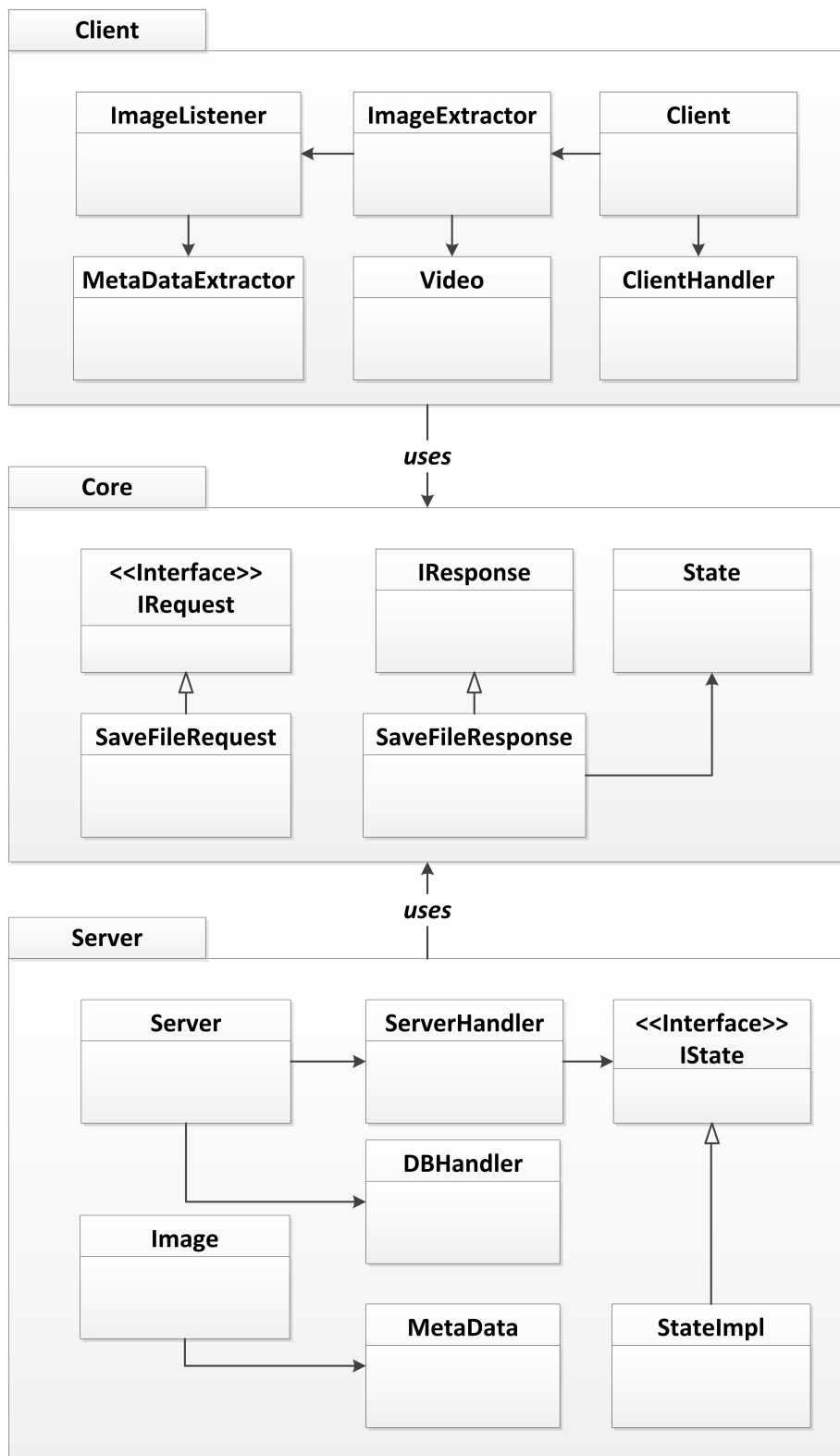


Figure 5.6: Class Diagram for Netty Client-Server application

Class diagram of client and server modules of our show case is shown in Figure 5.6. We have separated client and server modules in three components client, server and core. Core contains all common features that can be used by both client and server. So, both of them use core component. Figure 5.6 shows classes of client, server and core components and their classes and interfaces.

5.3 Computation Scenarios

- *Computation in Camera*: Retrieving images and metadata in individual cameras and sending to the Cloud.
- *Computation in Gateway*: Connecting all Cameras to Gateway, retrieving images and corresponding metadata from all streaming videos in Gateway, and sending images and corresponding metadata to the Cloud.
- *Computation in the Cloud*: Sending videos to the Cloud and retrieving images and corresponding metadata in the Cloud.

We have tested our application for the second scenario. These scenarios and deployment in real cloud environment helps us play around Critical Infrastructure and the Cloud. Based on our experience while developing the show case and a literature review, we identify security and privilege issues in CI and the Cloud. We are listing the identified security issues in section 5.4.

5.4 Security Issues of CI in the Cloud and Anticipated Result

Data Security

Data security refers to protective digital privacy measures that are applied to prevent unauthorized access to computers, databases and websites.

Data Storage

Issue: Data of cloud applications are stored in the Cloud storage server. Users access and update data frequently, so correctness in dynamic data update has high importance. Data storage protection methods e.g., redaction, truncation, obfuscation has to be considered in the Cloud [53–55]. Correctness and availability of data must be guaranteed, which influence the Quality of Service. Service Level Agreement should state trust between the providers and clients focusing on the utilities consuming behavior in the Cloud [56]. Data Storage Security issue is a technical threat. This threat is caused if cloud applications are not able to meet availability and confidentiality security objectives of information.

Example from show case: Videos can be of different types, for example streaming videos or stored videos. Somebody could manipulate videos stored in database. If a video is changed then information given by the video is also changed. This issue is possible in video storing database or database in the cloud where images and corresponding metadata are stored. If an attacker is

able to add information in metadata and add to the database in the Cloud then it increases the consumption power. As a consequence, utility metering in the Cloud automatically increases pay-per-use cost.

Anticipated Result: We want to apply software security standards, guidelines or tools and find out how to make database storage secured for example, detect changed data, restore the original video or images and metadata, and provide correctness of data.

Data Breaches or leakage

Issue: Sensitive internal data of an organization can fall into wrong hands while storing, transferring, processing or auditing. It is possible by different means. If a multi-tenant cloud service database is not properly designed, a flaw in one client's application could allow an attacker access not only to that client's data but to every other client's data as well. Offline backups of data to reduce impact of catastrophic data loss also increase data breaches. Interception of data between the customer and the Cloud provider leads to data leakage to third party [45, 57, 58]. Data Breaches or leakage issue is a technical threat created due to the violation of confidential security objective.

Example from show case: Somebody not in an organization get access to celebrity video or intercept video, image or metadata. He/she can misuse information in the video or give it to competitive organization. It could happen in various ways depending on the computation scenarios in section 5. For first scenario i) video can be accessed in camera which has captured video ii) images and corresponding metadata could be intercepted while transferring from camera to the Cloud. For second scenario i) video can be accessed in gateway where all available videos are collected for computation ii) videos could be intercepted while moving from camera to Gateway iii) images and corresponding metadata could be intercepted while transmitting to the Cloud. For third scenario i) video can be accessed in Cloud if the video is sent to Cloud for computation ii) video could be intercepted while transferring from Camera to the Cloud.

Somebody not in an organization get access to celebrity video. He/she can misuse information in the video or give it to competitive organization. Video can be accessed in i) database if the video is stored in database, ii) camera which has captured video, iii) gateway where all available videos are collected for computation iv) Cloud if the video is sent to the Cloud for computation.

Anticipated Result: After applying a software security standards and tools, we want to detect who access which videos and which videos or images and metadatas are intercepted. We want to implement a mechanism to trace the interception.

Data Loss

Issue: Data stored in the Cloud can be accessed and erased by attackers or accidentally. Physical catastrophe such as fire or earthquake also lead to permanent data loss. Thus, cloud providers should take adequate measures to backup data. If a customer uploads encrypted data in the Cloud and loses the encryption key then in this case also data will be lost [57]. Data Loss is a technical threat violating the availability security objective.

Example from show case: There are two cases where data loss could happen. One is someone deliberately delete videos from database or cameras and the other is images and metadata are lost from the Cloud storage.

Anticipated Result: After applying software security standards and tools, we expect to restore the deleted data or detection of loss data. To apply this, we want to implement data loss prevention approaches.

Data Scavenging

Issue: Data cannot be completely removed from storing devices until the device is destroyed. Data could be recovered from the hardware or using other techniques [58]. Attackers take this opportunity to recover the critical data. Data Scavenging is a technical threat and it causes the violation of confidentiality.

Example from show case: Although we remove videos or images and metadata from databases or from Camera by keeping security in mind, attackers could recover the data.

Anticipated Result: We want secure disposal of data storing devices. After applying the appropriate software security standards and tools, we want to develop a technique or a mechanism for complete removal of data from the storing devices.

Transmission Security or Network Security

Issue: Cloud providers transfer data to their client through internet. Due to the possibilities of sniffing, spoofing, hijacking and many more threats available in the Internet, data encryption and communication in secure channel have big space in security [53–55]. One distinctive aspect from traditional IT system is CI built on very simple network device for example SCADA. SCADA was historically isolated from systems protected by security mechanisms and provides remote coordination of control systems. Integration of CI in the Cloud requires secure transmission. Data Transmission Security is a technical threat that occurs due to the violation of confidentiality of information security.

Malicious Virtual Machine (VM) can listen to virtual network (VPN, VLAN) and even use ARP spoofing to redirect packets to/from other VMs [58]. Thus, there is a threat of sniffing/spoofing virtual networks. Virtual Network security issue is a technical threat that causes problem of confidentiality.

Example from show case: In our show case scenarios, either video or image and corresponding metadata are transmitted through internet to the Cloud. TCP Spoofing could result different images and metadata. Attackers could misuse the information getting from sniffing the transmission. There are different possibilities of connecting client and server for our show case. If we use virtual network for the connection and attackers listen this virtual network then they are able to access requests sent by client with critical data (images and metadata).

Anticipated Result: Applying software security standards and tools, we want to prevent such attacks in the Internet. To prevent attacks, we want to authenticate the external connections, control network connection and many more so that it helps to make secure connection, ports over internet.

Application Security

Issue: Applications are protected from threats using software, hardware and procedural methods. Security built into applications minimizes hackers attacks. Consideration of security during development is very important. Application level requires security such as service availability, software security, communication, access control, data protection security. Attackers are turning their attention to the common weaknesses created by application developers and threats such as Cross site scripting, Malicious file execution, Injection flaws attacks which are possible in the Internet [54, 55]. Open Web Application Security Project (OWASP) [59] is dedicated to help organizations understand and improve the security of their web applications and web services. Consideration of countermeasures to the possible attacks helps to mitigate the attacks. CI applications are often not even built for being exposed to public internet but potentially share infrastructure in the Cloud. Application Security issue is a technical threat which can be caused due to weaknesses of confidentiality or integrity and can cause problem of availability.

Example from show case: In our show case injecting malicious code in search web application attackers can modify data, delete data, drop table etc. in the database in the Cloud. Attackers can inject malicious code in the input field of web application available for searching images by metadata. We have considered good coding style like writing codes and testing, and have followed test-driven-development in the show case.

Anticipated Result: After applying security standards and tools, we want built-in security and security such as service availability, communication, access control and data protection in application level. Additionally, we want to prevent such attacks possible in the Internet.

Virtual Machine Security

Creation Security

Issue: An attacker who has valid account can create VM containing malicious code such a Trojan horse and store in the provider's repository [58]. VM created in the Cloud on which applications are supposed to be run and should not contain any malicious code. When virtual machines run overtime and cannot be managed by administrator then it is called VM Sprawl. VM Sprawl consumes overuse of infrastructure and increase cost. It is due to the creation of VMs without proper procedures or control of the release of these VMs. Virtual Machine Creation Security issue is a technical threat that violates multiparty trust security objective.

Example from show case: We have created VM in VMWare to deploy our web application and server. If the VM we have created contain any malicious code then it may change files which affect Server and web application or shutdown VM. As a consequence our show case will not be available any more.

Anticipated Result: Applying software security standards and tools, we want to detect and avoid any malicious code while creating VM in the Cloud. We want to perform the actions in secure manner. Thus, we want a mechanism that avoids or detects malicious code.

Isolation Security

Issue: Several VMs run in a common platform. If VMs in a platform are not strong isolated, a user of a VM can access other VMs and can access sensitive information [30]. VM hopping threat happens when a VM is able to gain access to another VMs (e.g., by exploiting some hypervisor vulnerability) [58]. Failure of separating storage, memory and routing effectively causes isolation failure [45]. Virtual Machine Isolation Security issue is also a technical threat. Occurrence of this issues exposes the confidentiality.

Example from show case: VM for our show case is running in VMWare. If a user of other VMs in the same platform access VM running our web application and Server, then the user can access data stored in database.

Anticipated Result: We expect that the application of software security standards and tools support strong isolation of VMs in same platform. As a result, a user of one VM cannot access other VM. If it is accessed then detect the user who access our VM.

Execution Security

Issue: VM execution environment must be secured to process sensitive data and establish connection with other party [30]. Assurance of execution environment should be provided while communicating messages across. Virtual Machine Execution Security issue is a technical threat that occurs violating multiparty trust.

Example from show case: Exchanging messages within VM environment, for example request from Client, saving data and sending response to Client by Server should be secured. If the communication message across environments is interrupted then required operations are not performed.

Anticipated Result: If we apply software security standards and tools in our show case, we want to process request of saving data, response to Client, saving and retrieving data.

Migration Security

Issue: Live migration of VMs expose content of VM state files to the Internet. Attacker can access data, transfer VM to an untrusted host, create and migrate several VMs causing disruptions or DoS [58]. Virtual Machine Migration Security issue is a technical threat in the Cloud. Live migration of VMs opens holes in confidentiality of VMs' information.

Example from show case: If attackers access images and metadata in the Cloud, then they can misuse, delete or update data. If there are requests from Clients in live migrations, attackers can mishandle these requests. Not only this, attackers could also transfer the VM to untrusted host.

Anticipated Result: After applying security standards and tools, we prevent exposing VM state files to internet, detect hosts to which VM is transferred to and detect who access the data.

Insiders Security

Issue: Insiders may misuse their privileges to access the Cloud intentionally or accidentally [30, 45]. For example Cloud System administrator can delete virtual machines for an organization.

Cloud hardware supplier can copy VM images of some organization and sell it to a competitor organization, invalidates backups. Lack of transparency of the provider process, access to virtual assets by employees, and lack of visibility of employees' roles and responsibilities cause several issues [31]. An employee or a user performs operation which is not allowed and lack of transparency hide who performs and completes such actions. A malicious threat to an organization can be a current or former employee, contractor or other business partner who has or had authorized access to an organization and intentionally exceeded or misused access in a manner that negatively affect confidentiality, integrity or availability of information [57]. Insiders Security issue is a technical and an organizational threat causes due to lack of mutual auditability of internal environment in the Cloud.

Example from show case: If VMWare system administrator deletes the VM where our application is deployed then our system will not be available anymore. If metadata are modified by administrator then searching facility have negative effect displaying incorrect information in the web.

Anticipated Result: After applying software security standards and tools, we want transparency of actions performed and completed by users and visibility of roles and responsibilities of users who have authorized access to organization's critical information. We want to restore or detect the deleted VMs and detect the users.

Interfaces and APIs Security

Issue: Cloud expose interfaces for management and interaction of cloud services. These exposed interfaces need to be standard and secured. Security and availability of services depend on the security of these basic APIs [57]. Thus, Cloud service provider depends upon APIs to deliver services to their customers. So, APIs must have secure authentication, encryption, activity monitoring mechanisms and access control [31]. Interfaces and APIs Security issue is a technical threat which can be caused due to problems in usability or providing mutual auditability or mentioning confidentiality.

Example from show case: If Interfaces for requests sent by client and responses sent by server are not standard and do not encapsulate information then critical information is exposed to breaches available in internet.

Anticipated Result: If we apply software security standards and tools, we expect security of these interfaces and detection of the exposed critical data.

User-Centric Security

Issues: Cloud providers have full control over the hosted services in Infrastructures. For example, they can control credentials to access virtual machines and servers where user data can be hosted. Users have very limited control over the service deployment and have no control over the exact location of provided service. Though users do not have control, they trust the guarantees provided in SLA [30]. As a consequence, regulation issues arise while moving or processing data across territorial boundaries. User-Centric Security issue is a violation of mutltiparty trust security and is an organizational threat.

Example from show case: While transferring data across territorial boundaries if regulations like “data protection” laws are not satisfied then data are not secured.

Anticipated Result: After applying software security standards and tools, we want to fix the regulations issues.

Shared Resources Security

Issue: Cloud Service Providers - CSPs provide scalable services by sharing infrastructure, platforms and applications [57]. Cloud Computing use same infrastructure used in the Internet and shared among cloud customers. Therefore, all current problems faced by the infrastructures are migrated in the Cloud. It is due to the reason that most of components are not designed for sharing resources in the Cloud [31]. Components (e.g., CPU caches, GPU, etc.) of shared infrastructure are not designed to offer strong isolation for IaaS, PaaS, SaaS. Single failure or misconfiguration can lead across an entire provider’s cloud [57]. VMs in same server share CPU, memory, I/O and other resources which decrease security of each VM. A malicious VM can infer information of other VMs through shared memory or other shared resources without compromising the hypervisor [58]. Therefore, defensive strategy including compute, storage, network, application is required. Shared Resources Security issue is a technical threat related to the usability of the cloud environment and cloud resources or applications.

Example from show case: Many VMs are sharing same platform and components. Since these components are not well-isolated, critical information is exposed to the unauthorized user. If critical data (images and metadata) in a VM is accessed on other VM then information could be leaked to competitors or attackers.

Anticipated Result: We want to detect data accessed through shared resources after implementing software security standards and tools.

We could not link some of the security issues to our show case but we could link them to other CI applications. Those issues are as follows:

Cloud Integrity Security

Issue: Data in the Cloud requires to be backuped in a regular basis to be safe from data-loss incident. When a user requests a service implementation then the Cloud system determines a free-to-use instance of the requested type and address for accessing the new instance and communicate it back to the user. For this identification purpose, it requires metadata on service implementation module. These metadata should be stored outside the Cloud to maintain the correct association of service implementation instances and metadata [54]. The CI metadata is important for billing and provision of basic needs but it also allows identifying the behavioral patterns. These critical information could be misused, so it needs to be secured. Cloud Integrity Security issue is a technical threat about the integrity of information.

Anticipated Result: After applying software security standards and tools, we want to secure the metadata used to determine the free-to-use instance and integrity of service implementation instance and metadata.

Security Related to Third Party

Issue: Trusted Third Party (TTP) [53, 54] establishes secure interaction with two parties (both of them trust third party) and provides end-to-end security services (based on security standards and tools) within the Cloud. All critical transactions between the two parties are reviewed by the Third Party. Data held by the Third Party is complex and has a lack of control and transparency. Therefore, Third party requires security on confidentiality, client and server authentication, certificate-based authorization and creation of security domain. Security Related to Third Party issue is a mutual auditability threat that creates weakness in multiparty trust.

Anticipated Result: After applying software security standards and tools, we want to trace actions of third party and support transparency that helps us to control third party.

Harmonization of security policies between Cloud layers and cloud providers

Issue: Self-automated services for a component depend on layer or sub-layer of Cloud Infrastructure. Agreement of the policy that governs interaction of such layer with other layers (i.e. properties and relation of the layer with other layers are to be considered) [30]. Harmonization of security policies between Cloud layers and cloud providers security issue is a technical threat. This issue occurs due to the violation of multiparty trust security in the Cloud.

Anticipated Result: Applying software security standards and tools, we want to make the security policies agreed by different layers for interaction.

Hypervisor Security

Issue: Hypervisor is a piece of software or hardware that creates and runs multiple VMs. Cloud administrator having access to hypervisor for servers can access memory of VMs which are able to access VMs [30]. VM escape threat is designed to exploit the hypervisor in order to take control of the underlying infrastructure [58]. Due to the possibility of such threats hypervisor access should be controlled and must be secured. Hypervisor Security issue is a technical threat and this issue is about the confidentiality of critical information in a VM.

Anticipated Result: We want to develop a mechanism to control and trace hypervisor access after applying software security standards and tools.

Account security

Issue: Hijacking account happens by social engineering or weak credentials [58]. By accessing user's credentials attackers can access sensitive data, manipulate data and redirect to any transaction. Attack methods such as phishing, fraud and exploitation of software vulnerabilities are used to get credentials and passwords [57]. After getting credentials attackers eavesdrop real user's activities and transactions, manipulate data, return falsified information and redirects clients to illegitimate sites. Account security issue is a technical threat. It is an issue related to confidentiality of credentials.

Anticipated Result: After applying software security standards and tools, we want to detect who access sensitive data, and trace the actions performed by attackers. So, we want to develop a mechanism for the detection.

Cloud Service Security

Issue: Attackers consume more resources which cause DoS or DDoS. People cannot access service or system slowdown when Dos or DDos occurs. Attackers as users misuse Cloud's benefit and use for attacking purposes. For example, it takes years to crack an encryption key using limited hardware but using an array of cloud servers, it might be possible to crack in some minutes. Attackers also stage a DDoS attack, distribute pirated software using array of cloud servers. Attackers access critical areas of deployed cloud computing services with stolen credentials and compromise confidentiality, integrity and availability of the services [57]. Service hijacking threat could happen when attackers hack a web site hosted in a Cloud Service Provider and install their software and control the Cloud Provider infrastructure [31]. Cloud Service Security issue is a contractual threat. This issue is also related to confidentiality security objectives.

Anticipated Result: After applying software security standards and tools, we want to make system that defenses (D)Dos attack. Therefore, we also want to develop defense approaches for these attacks.

Categorization and Classification of Secure Software Development Means

Many software security standards, guidelines and tools are available to develop secure software. Implementing these software security standards and guidelines helps us to ensure the security of a software. These available popular secure software development means are used in different context. In Chapter 5, we have identified a list of security issues in CI context in the Cloud. We have to implement applicable security standards, guidelines or tools to fix these issues. In this Chapter, we classify and categorize these security standards, guidelines and tools in different dimensions.

6.1 Classification and Categorization

In this section, we classify and categorize the software development means according to their applicability. We will classify these secure software development means to *i*) Standards and *ii*) Guidelines and Tools. Classification of these means are summarized in a table at the end of this section.

Standards

A standard is a document that provides requirements, specifications, guidelines or characteristics that can be used consistently to ensure that products, processes and services are standardized and fit to their purpose. Standards that provide techniques to implement security to minimize attacks in an application are called security standards. Security standards that provides security techniques to minimize the number of cybersecurity attacks are known as cybersecurity standards. Standards are strick and do not change often and certification of these standards also helps to get assurance as well. There are different types of international standards available. We are discussing these types in the following sections.

ISO Standards

International Organization for Standardization - ISO is an international standard representing various national standards organizations. ISO International Standards ensure that products and services are safe, reliable and of good quality. ISO standards are strategic tools that reduce costs by minimizing waste and errors, and increase productivity. They help companies to access new markets. ISO has published a list of ISO-Standards for different purposes. Some of the popular ISO standards are ISO 27001-Information management security, ISO 27002-code of practice-information security, ISO 31001-Risk management. We are considering some of the popular standards in our survey.

IEC Standards

International Electrotechnical Commission - IEC is a non-profit, non-governmental international standards organization. IEC prepares and publishes International Standards for electrical, electronic and related technologies - known as “electrotechnology”. IEC standards cover technologies from power generation, transmission and distribution to home appliances and office equipments, semiconductors, fibre optics, batteries, solar energy, nanotechnology and many others. IEC has also provided a list of IEC standards.

ISO has formed joint committees with IEC to develop standards and terminology in the areas of electrical, electronic and related technologies. These standards are known as ISO/IEC standards. For instance, *ISO/IEC 27002 - Security techniques – Code of practice for information security management* is an information security standard published jointly by ISO and IEC. Some of the popular ISO/IEC standards are as follows:

- *ISO/IEC 27001:2005- Information technology Security techniques - Information security management systems - Requirements*: ISO/IEC 27001:2005 [16, 17] was published by International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) in October 2005. It is the specification for establishing, implementing, operating, monitoring, reviewing, maintaining and improving a documented Information Security Management System. It ensures the protection of information assets and give confidence to interested parties.

Certification to ISO 27001 allows to prove managing the security of information to clients and other stakeholders. It provides a framework for the management of information security. It allows continuous management security policies and procedures that reduce information security risk on an ongoing basis [60].

- *ISO/IEC 27002:2005- Information Technology - Security Techniques - Code of practice for information security management*: ISO/IEC 27002:2005 [18] was published by International Organization for Standardization - ISO and the International Electrotechnical Commission - IEC in 2005. It provides best practices for initiating, implementing or maintaining information security management systems. It suggests a set of controls to address information security risks, covering confidentiality, integrity and availability aspects.

ISO/IEC 27002:2005 also provides a code of practice for information security professionals. To achieve a consistent and reliable security program (software), many organizations have adopted the ISO 27002:2005 standard as a key compliance strategy [61].

NERC Standards

North American Electric Reliability Corporation - NERC was formed in 2006 and it is a non-profit corporation. Mission of NERC is to ensure the reliability of North American bulk power system. So, NERC develops standards for power system operation, monitoring and enforcing compliance with those standards. It not only provides standards but also investigates and analyzes cause of disturbances in power system to prevent future events.

Concept relating to the preparedness and response to serious incidents that involve the Critical Infrastructure is called as Critical Infrastructure Protection - CIP. NERC also provides standards for CIP which are called NERC CIP.

- *NERC CIP*: NERC CIP [62] provides a set of cyber security standards through CIP-002 to CIP-009 and sets a baseline for CI cyber security. Most widely recognized standard NERC 1300 (update of NERC 1200) is from CIP-002-3 to CIP-009-3. The NERC CIP standards are the following:

1. CIP-001 Sabotage Reporting
2. CIP-002 Critical Cyber Asset Identification
3. CIP-003 Security Management Controls
4. CIP-004 Personnel and Training
5. CIP-005 Electronic Security Perimeter(s)
6. CIP-006 Physical Security of Critical Cyber Assets
7. CIP-007 Systems Security Management
8. CIP-008 Incident Reporting and Response Planning
9. CIP-009 Recovery Plans for Critical Cyber Assets

Common Criteria

Canada, France, Germany, Netherlands, United Kingdom and United States developed the security evaluation standard “Common Criteria for Information Technology Security Evaluation” in 1996. It is often referred to “Common Criteria”- CC. CC is useful as a guide for the development, evaluation and/or procurement of IT products with security functionality. CC provides security functional requirements and different evaluation levels that help to develop secure software. Providing security requirements at the early stages of software development in a systematic way and integrating Common Criteria into the software lifecycle develops the concept of security engineering [63]. CC [10] is an ISO/IEC 15408 international standard and has provided its documentation in three sections. Introduction and general model section includes the purpose, and

principles of security evaluation, and model of evaluation. Security functional components section includes a set of security functional requirements that are expressed in a Protection Profile - PP or a Security Target - ST. PP defines a standard, a set of security (including functional and assurance) requirements for a product. ST defines the security properties of the product to be evaluated. It may refer to one or more PPs. These requirements describe behaviour of target of Evaluation and intend to meet the security objectives in PP or ST. Security assurance components includes security assurance requirements, various methods to assure a secure product. This third section also includes seven Evaluation Assurance Levels - EALs, defining a scale for measuring assurance. They are the following:

- EAL1 (Functionally tested),
- EAL2 (Structurally tested),
- EAL3 (Methodically tested and checked),
- EAL4 (Methodically designed, tested and reviewed),
- EAL5 (Semi-formally designed and tested),
- EAL6 (Semi-formally verified design and tested ISO/IEC 2700),
- EAL7 (Formally Verified design and tested).

Security functional and assurance requirements are specified by users through PP, implement and evaluate security attributes of the products. A PP identifies the desired security requirements of a product and can include both functional and assurance requirements of the product type. This process provides the assurance of specification, implementation and evaluation. CC aims to eliminate redundant evaluation activities, reduce/eliminate activities that contribute little to the final assurance of a product, clarify CC terminology to reduce misunderstanding, restructure and refocus the evaluation activities to those areas where security assurance is gained.

Security Development Lifecycle

Security Development Lifecycle - SDL [6,7] is a software development process that helps to develop secure software addressing security. Since 2004 SDL has embedded security and privacy in software. SDL receives several international certificates: EN ISO 9001 (quality management system), EN ISO 15038 (translation services), EN ISO 13485 (medical devices quality management systems), EN ISO 14001 (environment management systems), EN ISO 27001 (information security management systems). Among them ISO/IEC 27001:2005 is one of the most successful international certifications held by SDL [64]. SDL introduces security and privacy in every phases of software development process and reduces vulnerabilities in software.

Microsoft SDL- MSDL is a secure software development process proposed by Microsoft. Ongoing education and training in software development is critical. Good knowledge transfer helps to react to changes in technology in a group. Regular evaluation of SDL processes

and changes to new technology helps to understand the cause and effect of security vulnerabilities. Microsoft has practiced SDL in different points in software development process and has applied security proven practices in Microsoft products that are running in different operating systems and platforms. Good security development practices and implementation priorities help to integrate secure development concepts in an existing development process. Microsoft SDL optimization model has addressed this issue. It is structured in five capabilities areas that roughly correspond to software development life cycle phases, and four maturity levels for the practices and capabilities in the areas. The five areas are i) Training, policy and organizational capabilities, ii) Requirements and design, iii) Implementation, iv) Verification, v) Release and response. The four maturity models are i) Basic, ii) Standardized, iii) Advanced and iv) Dynamic. MSDL provides software development tools for different phases of secure software development [8].

Guidelines and Tools

Guideline is a document that provides best practices that help to secure IT products. Guidelines also provide the security requirements, specifications that can be used to ensure products. Guidelines and standards both provide documents for different purposes including security but guidelines are loose and are changing frequently. It is important that cloud providers should take appropriate security measures. These measures should be based on best-practices. Security is constantly changing and security measures must be improved continuously. Thus, best practices are changing continuously considering the improvement. It is important to prevent and mitigate the impact of cyber attacks by creating logical redundancy. That is, defense attacks using different layers and separating systems with a different logical structure, cross-check transactions and detect attacks. All these activities are best practices considering cyber security.

Different tools are available that support different phases of software development. Here, we are considering tools that help us to develop secure software in the Cloud in CI context. These provided tools are configured in an application and feed parameters as input. Running the tools after feeding the parameters generate some output. These outputs depend on the tools how they show output. From this output we can benefit on developing secure software. Various tools are available for producing secure software. These tools are automated tools for verification and validation of formal specifications and design. For instance, Security Development Life cycle - SDL is providing a threat modeling tool called SDL Threat Modeling Tool. We are discussing some popular guidelines and tools as follows:

SDL Threat Modeling Tool

Threat modeling allows software architects to identify and mitigate potential security issues in early (design) phase of a software development. It increases awareness in threats, helps to focus resources, do better security assessments, choose the appropriate tools and implement the best design. Threat modeling is a core element of Microsoft SDL. SDL Threat Modeling Tool uses STRIDE and DREAD. STRIDE classifies threats in six categories i) Spoofing, ii) Tampering, iii) Repudiation, iv) Information disclosure, v) Denial of Service, and vi) Elevation of privilege. DREAD measures the threat risk level. Risk of a threat is measured by grading vulnerability in all of the following categories i) Damage, ii) Reproducibility, iii) Exploitability, iv) Affected

Users, and v) Discoverability. Risk calculation formula is: $Risk = (D + R + E + A + D)/5$. Threat modeling with STRIDE and DREAD helps software architects to design secure software by producing the best design.

Correct by Construction

Correct by construction - CbC [9] methodology is a high-integrity software development process especially for security- and safety-critical applications. CbC is developed by Praxis Critical System. It overcomes the uncertainty of security critical. CbC demands a software development process that builds correctness in every step. It requires rigorous requirements definition, precise specification, verifiable design, behavior understandable code, and defect prevention and removal techniques. Key principles of CbC are:

- expect requirements to change,
- know why you are testing,
- eliminate errors before testing,
- write software that is easy to verify,
- develop incrementally,
- some aspects of software development are just plain hard
- software is not useful by itself.

Computer Emergency Response Team

Computer Emergency Response Team - CERT [13] Coordination Center was created in 1998 and is located at Software Engineering Institute, Carnegie Mellon University, USA. It is recognised as trusted, authoritative organization dedicated to improving the security and resiliency of computer systems and networks. The CERT Program is a national asset of cybersecurity. CERT developed advanced methods and technologies to counter cyber threats like “Common Sense Guide to Mitigating Insider Threats” [14], OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) Methods [15]. Common Sense Guide to Mitigating Insider Threats provides

- the most recommendations from the CERT program based on database of more than 700 insider threat cases and
- 19 best practices and details how and why to implement them.

OCTAVE Method is a three-phased approach. It also provides OCTAVE method implementation guide. The three phases are as follows:

- phase 1: Build Assets-Based-threat profiles

- phase 2: Identify Infrastructure vulnerabilities and
- phase 3: Develop Security Strategy and Plans.

CERT provides secure coding standards for commonly used programming languages such as C, C++, Java, and Perl. It also provides language independent coding best practices [65].

European Network and Information Security Agency

European Network and Information Security Agency - ENISA is an agency of European Union, created in 2004 by EU. ENISA addresses network and information security problems and advises best practices to prevent issues. Authors in [11, 12] provide ENISA security monitoring framework guide. It gives guidance to monitor security service levels and governance of out-sourced cloud services. Its Cloud Computing Risk Assessment report provides assurance criteria to assess the risk of adopting cloud services. ENISA provides guidelines to develop a secure smart phone with OWASP. It means, it provides guidelines to develop an application applying countermeasures of possible attacks [66]. Classification of these secure software development means is shown in table 6.1. These means supported by the Standards or Guidelines and Tools are marked in the Table.

Means	Standards	Guidelines and Tools
SDL/MSDL	✓	
ENISA guidelines		✓
CERT best practices		✓
ISO27001	✓	
ISO27002	✓	
CC	✓	
CbC		✓
NERC CIP	✓	

Table 6.1: Categorization of popular standards, guidelines and tools studied in section 6.1 in standards, and guidelines and tools.

6.2 Sub-categorization

Popular secure software development means were classified and categorized in the previous section. In this section, we are going to sub-categorize them into two dimensions, firstly we will sub-categorize the Standards in i) security and ii) software development, secondly we will sub-categorize similarly for Guidelines and Tools. We will summarize this result in a table at the end of this section.

Security

Security standards facilitate the implementation of security controls and most of the time are used in the context of information security policies. Information security policies are high-level statements or rules about protecting systems. Security standard is a low-level prescription company that can enforce the given policy. Thus, security standards help to built-in security in an application as well as in operating environment. There are different types of standards supporting to develop secure software. We are sub categorizing standards, guidelines and tools discussed in the previous section depending on their way of producing secure software. If the standard, guidelines or tools help to make software secure by enforcing security policies or help to make more secure against the attacks by other means then we list them in this category. Following is the list of secure development means that falls in this category:

- ISO/IEC 27001
- ISO/IEC 27002
- NERC CIP
- CERT best practices

Software Development

Software development standards define framework for software life cycle processes, contain a hierarchy of processes, activities and tasks to be applied in software development environment. Thus, software development standards provide best practices and rules to be applied in different phases of software development life cycle (from requirement phase to deployment phase). Activities involved in software development from the provided standards or guidelines are applied to produce secure software. It reduces cost and reduces vulnerabilities. Not only best practices, there are also tools available that help in secure software development. Similar to standards and guidelines, these tools also help in different phases of software development life cycle. Thus, different tools are to be used for different purpose. The popular secure software development means we have discussed in the previous section that fall in this category are as follows:

- SDL/MSDL
- ENISA guidelines
- CC
- CbC
- SDL threat modeling

Sub-categorization of these secure software development means is shown in Table 6.2. These means supported by different categories are marked in the Table

Means	Security Standards	Software Development Standards	Security Guidelines and Tools	Software Development Guidelines and Tools
SDL/MSDL		✓		
SDL threat modeling				✓
ENISA guidelines				✓
CERT best practices			✓	
ISO27001	✓			
ISO27002	✓			
CC		✓		
CbC				✓
NERC CIP	✓			

Table 6.2: Sub-Categorization of popular secure software development means categorized in section 6.1 into security and software development.

Mapping of Security Issues and Popular Secure Software Development and Security Engineering Means

We have a list of identified cloud security issues on one hand and a list of popular secure software development standards, guidelines and tools on the other hand. For each of the identified issues in Chapter 5 section 5.4, we find out applicability of a representative subset of the popular secure software development standards, guidelines and tools discussed in Chapter 6. At first, we find out how these means help in addressing the issues and then we tick the boxes of the means and issues in a section as a matrix. All of the security issues are not addressed by a particular software standard, guideline or tool. In this case, we cross the boxes. For each of the ticked boxes we give reason how the particular mean helps to address the issue, and for the not addressing issues (crossed boxes) we also give reason in a summary.

7.1 Contribution of Security Means

SDL/MSDL

Microsoft SDL is a collection of mandatory security activities. These activities are grouped by the phases of software development life cycle (SDLC) and ordered in the order they should occur. Practical experience at Microsoft has shown that security activities are executed as a part of a software development process. To comply with the Microsoft SDL process, a development team should successfully complete sixteen mandatory activities. Security activities corresponding to phases of software development is shown in Figure 7.1. For example in design phase, establishing design requirements, analyzing attack surface and threat modeling are the security activities to be considered. Similarly, corresponding security activities are listed for other phases of software development in the Figure.

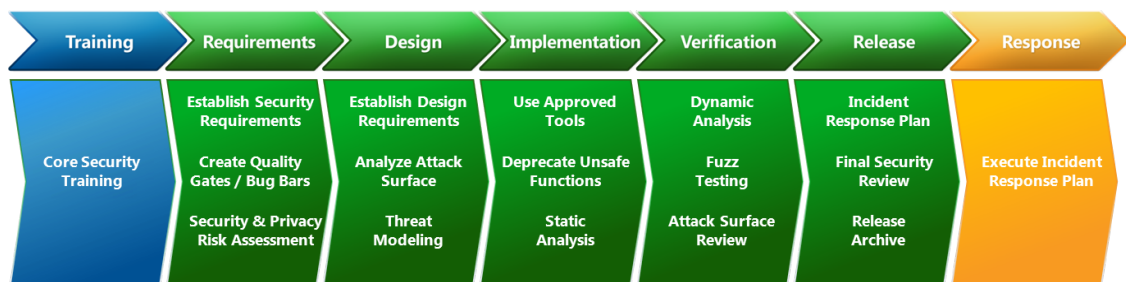


Figure 7.1: The Microsoft Security Development Lifecycle.[Source: Simplified Implementation of the SDL]

SDL/MSDL provides techniques to integrate security in every phases of software development process (requirement analysis, design, implementation, deployment) and helps to reduce vulnerabilities in software [7].

Application Security

In Practice 1, under topic *Training Requirements*, it is stated that security training of secure design, threat modeling, secure coding (buffer overflow, cross-site scripting, SQL injection) and security testing need to be given to software development team before the development of software. In practice 5 (*Design Requirements*), it is mentioned that secure features (well engineered functionality with respect to security) and security features (functionality with security implications like firewall) should be considered in the design phase. In Practice 7, *Threat Modeling* helps to consider security issues in application level and implication of security in a planned operational environment and structured fashion. *Static Analysis* (Practice 10) helps to ensure that secure coding polices are followed. *Dynamic Program Analysis* (Practice 11) ensures the correct functionality of the programs as they are designed.

Interfaces and API Security

In Practice 9, *Deprecate Unsafe Functions* states that project teams should analyze all functions and APIs that are used in software development projects and prohibit those determined to be unsafe, use code scanning tool to check code of banned functions and replace the banned functions with safer alternatives.

Data Storage Security

Optional security activities may be added at the discretion of the project team or the security adviser to achieve desired security objectives. One of the optional security activities is *Manual Code Review*. It is usually focused on “critical” components of an application. These critical components are the components where sensitive data like Personal Identifiable Information (PII) is processed and stored. So, it helps to have a correct-and-secure processing and storage of critical data.

ISO/IEC 27001

ISO/IEC 27001 is an internationally accepted standard for information security management. It is not only an IT standard but also a process, technology and people management standard. It adopts the Plan-Do-Check-Act (PDCA) model and helps to combat fraud and promote secure operation. It is an unified standard for security associated with the information life cycle. The specific security controls to implement per ISO 27001 are found in Annex A of the International Standard [67]. They are organized around the following numbered topics that correspond to the standard:

- A.5. Security Policy
- A.6. Organizing Information Security
- A.7. Asset Management
- A.8. Human Resources Security
- A.9. Physical and Environmental Security
- A.10. Communications and Operations Management
- A.11. Access Control
- A.12. Information Systems Acquisition, Development and Maintenance
- A.13. Information Security Incident Management
- A.14. Business Continuity Management
- A.15. Compliance

Security requirements engineering is an important step in “requirement phase” of software development. ISO/IEC 27001 provides security requirements which are to be considered in software implementation phase. For example: access control, communications and operations management, information system acquisition, development and maintenance in the development secure the system. It helps not only in engineering and developing secure software but also in securing Critical Infrastructure. Thus, we consider this standard as a secure software development standard focusing on Critical Infrastructure.

Transmission Security

Implementing ISO/IEC 27001 helps in securing the company information from being misused by unwanted intruders and assuring the overall safety of personnel and assets information. Section A.10 (Communication and operation management) [68] contains a sub-section for *Network security management*. Objective of this section is to ensure the protection of information in networks and protection of the supporting infrastructure. It provides *network controls* and *security of network services*. To ensure the security of electronic commerce services and their

secure use, ISO/IEC 27001 also provides requirements for securing electronic commerce, on-line transactions and publicly available information. Network access control requirements to prevent unauthorized access to networked services are the following:

- Policy on the use of network services
- User authentication for external connections
- Equipment identification in networks
- Remote diagnostic and configuration port protection
- Network connection control
- Network routing control

In Section A.15 (Compliance), *Regulation of cryptographic controls* is about Cryptographic controls that must/can be used in compliance with all relevant agreements, laws, and regulations. This helps in secure transmission.

Insiders Security

Section A.8 (Human Resource Security) [68] contains guidelines for dealing with personnel prior to employment, during employment and termination of change of employment. Objectives of these sections are as follows:

- *Prior to Employment*: The objective is to ensure that employees, contractors and third party users understand their roles and responsibilities. This reduces the risk of theft, fraud or misuse of facilities.
- *During employment*: All employees, contractors and third parties must be aware of information security threats and concerns, responsibilities and support organizational security policy in normal work. It reduces risks associated to human activities.
- *Termination or change of employment*: The objective is to ensure that employees, contractors and third party users exit an organization or change employment in an orderly manner.

Data Storage

In Section A.15 (Compliance), the sub-section *data protection and privacy of personal information* focuses on data protection and privacy, which must be ensured as required by relevant legislation, regulations, and, if applicable, contractual clauses.

Data Breaches or Leakage

The objective of Section A.15 (Compliance) is to avoid breaches of law, statutory, regulatory or contractual obligations and any security requirements. This is defined in sub-section *compliance with legal requirement*.

Data Loss

The objective of Section A.15 (*Protection of organizational records control*) is to protect important records from loss, destruction and falsification, in accordance to statutory, regulatory, contractual, and business requirements.

Security Related to Third Party

Section A.10 (Communications and operations management), the sub-section *Third party service delivery management* is about implementing and maintaining the appropriate level of information security and service delivery in line with third party service delivery agreements. Controls are discussed in the following sections:

- *Service delivery*: It is important to ensure that the security controls, service definitions and delivery levels included in the third party service delivery agreement are implemented, operated, and maintained by the third party.
- *Monitoring and review of third party services*: The services, reports and records provided by the third party shall be regularly monitored and reviewed, and audits shall be carried out regularly.
- *Managing changes to third party services*: Changes to the provision of services, including maintenance and improvement of existing information security policies, procedures and controls, shall be managed, taking into account the critical nature of business systems and processes involved and re-assessment of risks.

Harmonization of Security Policies

Objective of *compliance with security policies and standards, and technical compliance* is to ensure compliance of systems with organizational security policies and standards. Compliance with security policies and standards is about ensuring all security procedures are carried out correctly to achieve compliance with security policies and standards. Technical compliance checking section states that information systems shall be regularly checked for compliance with security implementation standards.

Application Security

Section A.10 (Communications and operations management), sub-section *Protection against malicious and mobile code* aims to protect the integrity of software and information. Controls are listed in the following subsections:

- *Controls against malicious code* : Detection, prevention, and recovery controls are required to protect against malicious code and appropriate user awareness procedures must be implemented.

- *Controls against mobile code*: Where the use of mobile code is authorized, the configuration must ensure that the authorized mobile code operates according to a clearly defined security policy, and unauthorized mobile code must be prevented from executing.

Account Security

The objectives of Section A.11 (Access Control) are defined in two sub-sections: *Business requirement for access control* and *User access management*. They ensure authorized user access to prevent unauthorized access to information systems. These are discussed in different sub sections:

- *Access control policy*: An access control policy must be established, documented, and reviewed based on business and security requirements for access
- *User registration*: There must be a formal user registration and de-registration procedure in place for granting and revoking access to all information systems and services.
- *Privilege management*: The allocation and use of privileges must be restricted and controlled
- *User password management*: The allocation of passwords must be controlled through a formal management process
- *Review of user access rights*: Management must review users' access rights at regular intervals using a formal process

ISO/IEC 27002

ISO/IEC 27002 is not a formal specification like ISO/IEC 27001. It is a code of practice/guideline - a generic advisory document as opposed to ISO/IEC 27001, which formally defines the mandatory requirements for an Information Security Management System (ISMS). ISO/IEC 27001 uses ISO/IEC 27002 to indicate suitable information security controls within the ISMS [69]. Typically, ISO/IEC 27001 and ISO/IEC 27002 are used together in practice.

ISO/IEC 27002:2005 comprises of ISO/IEC 17799:2005 and ISO/IEC 17799:2005/Cor.1:2007. Its technical content is identical to that of ISO/IEC 17799:2005. ISO/IEC 17799:2005/Cor.1:2007 changes the reference number of the standard from 17799 to 27002. The objectives of ISO 27002:2005 is to provide general guidance on the commonly accepted goals of information security management. It is summarized in 19 sections. After three introductory sections (framework, acceptable use, definition and terms), ISO/IEC 27002:2005 contains best practices of control objectives and controls in the following areas of information security management [70]:

- 4. Risk Assessment
- 5. Security Policy
- 6. Organization of information security

- 7. Asset management
- 8. Human resources security
- 9. Physical and environmental security
- 10. Communications and operations management
- 11. Access control
- 12. Information systems acquisition, development and maintenance
- 13. Information security incident management
- 14. Business continuity management
- 15. Compliance

Security is the main requirement of secure software system development and one of the most dominant issues for the success of a system and increases quality of software. Security intimidation affects the functionality and efficiency of software system, thus this intimidation needs to be considered while developing a system. For example, Nazir et.al. [71] propose a model based approach to evaluate security aspects of software components incorporating attributes of ISO/IEC 27002 standard and find out the most secure component in component based software development. These attributes are considered while developing a software. Therefore, we consider ISO/IEC 27002 as a software development development standard.

Application Security

Issues related to system development and maintenance are addressed in section 12 entitled *information systems acquisition, development and maintenance*. It is described as follows [72]:

- *Security requirements of information systems*: The aim is to ensure that security is built into IT systems. Therefore, an analysis of security requirements should be carried out at the requirement analysis phase of each development process.
- *Correct processing in applications*: The aim is to prevent loss, modification or misuse of user data in application systems. Controls like *input data validation*, *control of internal processing*, *message integrity* and *output data validation* should be performed to make sure that applications process information correctly.
- *Technical vulnerability management*: The aim is to reduce risk arising from exploitation of public technical vulnerabilities. Technical vulnerability management should be implemented in an effective and systematic way. These measurements should be confirmed to its effectiveness against the vulnerabilities.

Transmission Security

A subsection in *information systems acquisition, development and maintenance* is about data transmission in secured manner by *Cryptographic controls*. Its objective is to ensure that IT projects and support activities are conducted in a secure manner. It aims to protect the confidentiality, authenticity and integrity of information through cryptographic means. It is also stated that there should be a policy regarding the use of cryptographic controls. A subsection *Network security management* in Section 12 aims to ensure the safeguarding of information in networks and the protection of the supporting infrastructure. In order to achieve this, a range of network security controls are required and security of network services should be considered.

Data Storage

Section 15 (*Compliance with legal requirements*) aims to avoid breaches of any statutory, criminal or civil obligations and of any security requirements. A subsection on *Protection of organizational records* states that important records of an organization should be protected from loss, destruction and falsification. This focuses on storing files securely. In section 10, the subsection on *backup* aims to maintain the integrity and availability of IT services by regular data back-up. It is stated that back-up copies of essential business data and software should be regularly taken and should be stored securely well away from the actual site. These backup media should be regularly tested.

Data Leakage

A subsection on *Data protection and privacy of personal information* in *Compliance with legal requirements* states that applications handling personal data should comply with data protection legislation and principles. *Prevention of misuse of information processing facility* states that IT facilities should be used only for the authorized business purpose.

Data Scavenging

In Section 7 (*Asset management*), a subsection on *Media handling* aims to prevent damage to assets and interruptions to business activities. Another subsection *Disposal of media* in this section states that media should be disposed securely and safely when no longer required. Sensitive information of an organization could be leaked to outside person through carelessness in disposal of data storage medium. There should be formal procedures established for secure disposal of media to minimize this risk.

Insiders Security

The objective of the section *Information System audit considerations* is to maximize the effectiveness and minimize interference to/from the information systems audit process. These are addressed in subsections as follows:

- *Information systems audit controls*: Audit activities like checks on operational systems should be carefully planned and agreed to minimize the risk of disruptions to business process.
- *Protection of information system audit tools*: Information system and audit tools (e.g. software or data files) should be protected to prevent possible misuse or compromise.

Third Party Security

Section 6 (*Organization of Information Security*), subsection *External Parties* aims to maintain the security of the organization's information and information processing facilities that are accessed, processed, communicated to, or managed by external parties. A part of this subsection *Addressing Security in third party agreements* states that agreement with third parties should comply with all appropriate security requirements. This agreement includes accessing, processing, communicating, managing or processing organization's information. Subsection *Third party service delivery management* aims to implement and maintain the appropriate level of information security and service delivery in line with third party service delivery agreements. Appropriate controls for service delivery, monitoring and review of third party services, managing changes to third party services should be established.

Hypervisor Security

Section 10 (*Communications and operations management*) aims to detect unauthorized information processing activities. It states that appropriate control like audit logging, monitoring system use, protection of log information, administrator and operator logs, fault logging, clock synchronization for accurate recording should be performed in for monitoring activities. This helps in monitoring hypervisor and activities performed by user that has accessed the hypervisor.

Virtual Machine Execution Security

Section 12 (*information systems acquisition, development and maintenance*), contains some subsections addressing the issues of execution environment of software. They are as follows:

- *Security of system files*: The aim is to maintain the security of application system software and data. Appropriate controls like control of operational software, protection of system test data, access control to program source library should be taken into account. It helps to avoid exposure of sensitive data in system.
- *Security in development and support processes*: The aim is to maintain the security of application system software and information. It describes some controls like access control to program source library, technical review of applications after operating system changes, access control to program source library and information leakage.

Account Security

Section 11 (Access Control) addresses security issues related to account security in different subsections:

- *Business requirement for system access*: The aim is to control access to business information by documenting and defining an access control policy.
- *User access management*: The aim is to prevent unauthorized computer access. User registration, privilege management (privilege restricted and controlled), user password management (user password should be securely controlled), review of user access rights in regular interview should be performed.

Common Criteria (CC)

Functional security requirements of common criteria [73–75] are organized in an hierarchical manner as classes, families and components. Classes are group of families which focus on specific areas of security, group of components sharing security objectives are families and related set of security requirements are components. Classes of CC are the following:

- Security Audit : Logging
- Communication : Identification of parties, repudiation
- Cryptographic Support : Cryptography
- User Data Protection : Access control
- Identification and Authentication
- Security Management : System security and management
- Privacy
- Protection of the system security functions
- Resource Utilization : Utilization limitations
- System Access : Access/connection limits
- Trusted path/channels : Secure channels, sockets

Data Storage Security

The class *User data protection* contains families specifying requirements related to protecting user data. The family *User data protection* addresses user data within a target of evaluation (TOE), during import, export and storage as well as attributes directly related to user data. The family *Stored data integrity* provides requirements that address protection of user data while it is stored within containers controlled by TOE security functionality (TSF). Integrity errors may affect user data stored in memory, or in a storage device. Stored data integrity monitoring requires that the TSF monitor user data stored within containers controlled by the TSF for identified integrity errors. Stored data integrity monitoring and action adds the additional capability to the first component by allowing for actions to be taken as a result of an error detection.

Application Security

The TSF utilizes cryptographic functionality to satisfy several high-level security objectives. These include identification and authentication, non-repudiation, trusted channel and data separation. It is implemented when TOE implements cryptographic functions. Two families *cryptographic key management* and *cryptographic operation*, address the management aspects of cryptographic keys and the operational use of those cryptographic keys respectively.

Transmission and Network Security

Families in *Trusted path/channels* class provide requirements for a trusted communication path between users and the TSF, and for a trusted communication path between users and the TSF, and for a trusted communication channel between the TSF and other trusted IT products. A trusted channel is a communication channel that may be initiated by either side of the channel, and provide non-repudiation characteristics with respect to the identity of the sides of the channel. A trusted path provides a means for the users to perform functions through an assured direct interaction with the TSF. Trusted path is usually desired for user actions such as initial identification and/or authentication, but may also be desired at other times during a user's session.

Account Security

The class *TOE access* specifies the functional requirements for controlling the establishment of a user's session. One family in this class (*Limitation on scope of selectable attributes*) defines requirements to limit the scope of session security attributes that a user may select for a session. Another family (*Limitation on multiple concurrent sessions*) defines requirements to place limits on the number of concurrent sessions that belong to the same user. Family, *session locking and termination* defines requirements for the TSF to provide the capability for TSF (Target Security Functions)-initiated and user-initiated locking, unlocking, and termination of interactive sessions. *TOE access history* defines requirements for TSF to display to a user, upon successful session establishment, a history of successful and unsuccessful attempts to access the user's account. *TOE session establishment* defines requirements to deny a user permissions to establish a session with the TOE. Another family in the class (*Identification and authentication*) addresses the requirements for functions to establish and verify a claimed user identity. Identification and

authentication is required to ensure that users are associated with the proper security attributes (e.g. identity, groups, roles, security or integrity levels).

Cloud Service Security

The class *Resource Utilization* is divided into three families, *Fault Tolerance* provides protection against unavailability of capabilities caused by failure. The family *Priority of Service* ensures that the resources will be allocated to the more important or time-critical tasks and cannot be monopolised by lower priority tasks. The family *Resource Allocation* provides limits on the use of available resources, therefore preventing users from monopolising the resources.

Hypervisor Security

Common Criteria provides certification for KVM hypervisor [76]. KVM hypervisors have been validated to virtual machine runtime isolation.

7.2 Matrix of Supporting Means and Issues:

Table 7.1 summarizes the above result. Standards that help to mitigate security issues are ticked and non-significant standards are crossed in the boxes.

No.	Security Issues	Software Security Standards, Guidelines and Tools			
		SDL/MSDL	ISO/IEC 27001	ISO/IEC 27002	CC
1	Data Security				
	a) Data Storage	✓	✓	✓	✓
	b) Data Breaches or leakage	✗	✓	✓	✗
	c) Data Loss	✗	✓	✗	✗
	d) Data Scavenging	✗	✗	✓	✗
2	Transmission and Network Security	✗	✗	✓	✓
3	Application Security	✓	✓	✓	✓
4	Virtual Machine Security				
	a) Creation Security	✗	✗	✗	✗
	b) Isolation Security	✗	✗	✗	✗
	c) Execution Security	✗	✗	✓	✗
	d) Migration Security	✗	✗	✗	✗
5	Insiders Security Issue	✗	✓	✓	✗
6	Interfaces and API Security	✓	✗	✗	✗
7	User-Centric Security	✗	✗	✗	✗
8	Shared Resources Security	✗	✗	✗	✗
9	Cloud Integrity Security	✗	✗	✗	✗
10	Security Related to Third Party	✗	✓	✓	✗
11	Harmonization of security policies between Cloud layers and cloud providers	✗	✓	✗	✗
12	Hypervisor Security	✗	✗	✓	✓
13	Account Security	✗	✓	✓	✓
14	Cloud Service Security	✗	✗	✗	✓

Table 7.1: Result of our survey. Applicability of popular software security standards, guidelines and tools in the identified list of security issues.

7.3 Summary Explanation of not Significant Security Means that do not support Security Issues

In Section 7.1 and 7.2, we have crossed the boxes based on the applicability of secure software development means. We also have left the blank boxes whenever a particular secure software development standard, guideline or tool does not address all the security issues in the list. In this case, the considered secure software development mean does not significantly contribute to the issues. To adequately address the issues, it has to provide the supporting policies, guidelines or best practices that fulfills our anticipated result and helps in fixing the issues. To support the issues, we need at least common features that helps us to fix them. Based on this we summarize how these issues are **not** supported by standards, guidelines or tools as follows:

- *Data Breaches or leakage*: Policies or best practices or guidelines to detect information leakage or misuse of information are required to address this issue. These are missing in the SDL/MSDL and CC.
- *Data Loss*: Data loss prevention strategies need to be defined. It is missing in the SDI/MSDL, ISO/IEC 27002 and CC.
- *Data Scavenging*: Techniques for completely removing data from data storing device for example database are also required. This is not supported by SDL/MSDL, ISO/IEC 27001 and CC.
- *Transmission and Network Security*: To secure transmission and network, guidelines for securing the network by detecting the parameters of networks are required. This is not addressed in SDI/MSDL and ISO/27001.
- *Virtual Machine Security*: Techniques or practices that secure virtual machine need to be considered. Virtual machine creation, isolation, migration issues are not addressed in any of the standards surveyed in this research (i.e. in SDL/MSDL, ISO/IEC 27001, ISO/IEC 27002 and CC) but execution securing techniques are addressed only in ISO/IEC 27002.
- *Insiders Security Issue*: This is an organizational issue, privilege and roles need to be maintained, if this is not addressed in the policies, guidelines or best practices then this issue is open. This issue is open in SDL/MSDL and CC.
- *Interfaces and API Security*: Code of practice (techniques) for finding and removing for example deprecated functions need to be provided.
- *User-Centric Security*: Regulations like data protection laws need to be fulfilled. Different rules and regulations of different countries need to be satisfied while processing data. Such rules and regulations are not addressed in any of the surveyed standards.
- *Shared Resources Security*: Shared resources in cloud environment need to be traced by keeping track of the access of the shared resources. This issue is also not addressed by the standards we have surveyed.

- *Cloud Integrity Security*: Policies to protect and secure metadata of cloud services in the cloud environment need to be defined, none of the secure development means we have surveyed address this issue.
- *Security Related to Third Party*: Third party transactions need to be visible using the practices or techniques provided by the standard. It is not available in SDL/MSDL and CC.
- *Harmonization of security policies between Cloud layers and cloud providers*: The standards SDL/MSDL, ISO/IEC 27002 and CC do not provide guidelines on implementation mechanisms and policies between the layers and sub-layers of Cloud.
- *Hypervisor Security*: Controlling the access to the hypervisor and tracing the access is required for the securing hypervisor. These techniques are missing in SDL/MSDL and ISO/IEC 27001.
- *Account Security*: Strong user authentication in a software development supports defending account security. Authentication techniques are not mentioned in the practices.
- *Cloud Service Security*: Defending techniques of (D)Dos attack are not mentioned in the practices, policies and code of practice of SDI/MSDL, ISO/IEC 27001 and ISO/IEC 27002. So, they do not directly support Cloud Service Security.

7.4 Multidimensional Taxonomy

We have identified a list of issues from a literature review and described in Chapter 5, categorization and sub-categorization of standards, guidelines and tools are mentioned in Chapter 6 and mapping of these security issues and the secure software development means are mapped in this Chapter. All these categorization, sub-categorization, mappings are summarized in a multidimensional taxonomy. The multi-dimensions and the mapping is shown in Figure 7.2 ¹. Different colors are used in the mapping of the issues to the secure software development means only to avoid the confusions of overlapping lines.

¹NERC CIP, ENISA Guidelines, CbC, CERT Best Practices are left for future work.

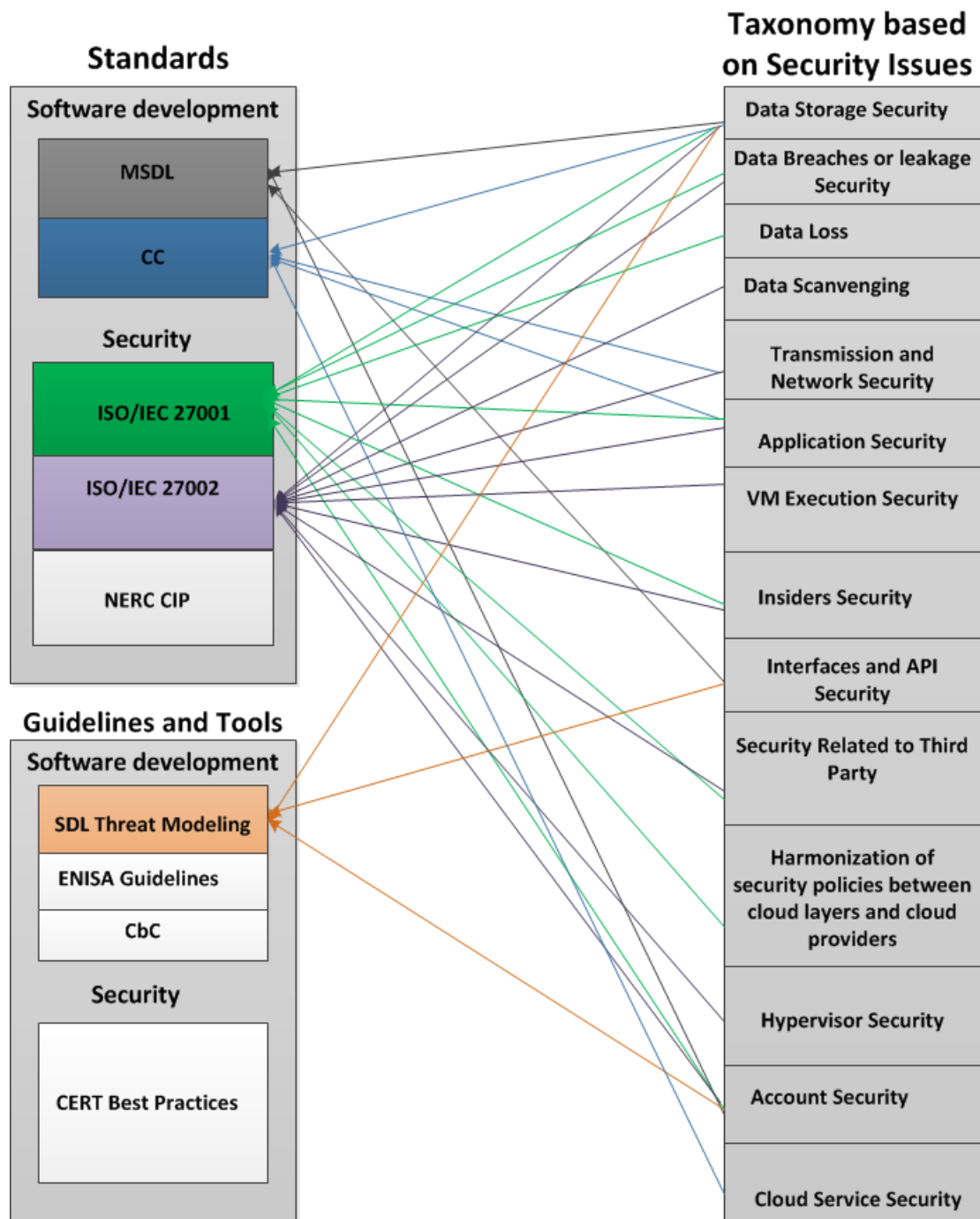


Figure 7.2: Multidimensional Taxonomy: Mapping the list of identified issues to the popular secure software developments means based on their applicability.

Evaluation of Taxonomy

We have mapping of security issues and secure software development means from Chapter 7. In this Chapter, we want to evaluate the mapping of the security issues to SDL/MSDL. For this, we describe principle of SDL Threat Modeling Tool, apply it in our show case, and do evaluation of mapping from the previous Chapter.

8.1 Evaluation Plan

In Chapter 5, we have identified a list of security issues from show case and literature analysis. We want to evaluate the mapping in Chapter 7 in this Chapter. Our evaluation plan is as shown in Figure 8.1. Issues from the show case are translated to security issues and evaluate the mapping selecting a tool from the standards in taxonomy.

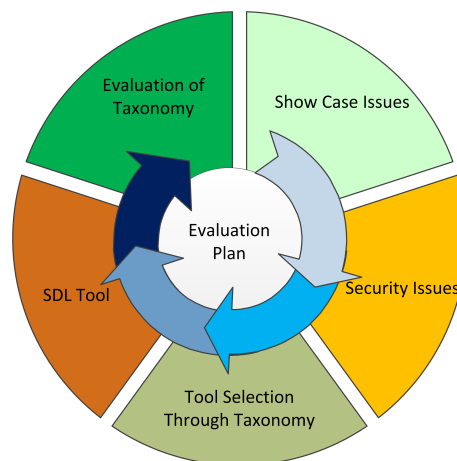


Figure 8.1: Evaluation Plan.

8.2 Security Issues from Show Case

Issues related to our show case and associated security issues are:

- Images and corresponding metadata are stored in elastic database in the Cloud. If an attacker is able to add information in metadata and add to the database in the Cloud then it increases the consumption power. As a consequence, utility metering in the Cloud automatically increases pay-per-use cost.

Associated security issue: Data Storage Security

- Depending on the computation scenario, video is available in different places. If unauthorized person get access to celebrity video either in database, camera, gateway or in Cloud. and misuse information in the video. The person could give it to competitive organization or misuse it.

Associated security issue: Data Breaches or leakage

- Videos can be deleted either from camera or database, and images and corresponding metadata can be deleted from the Cloud storage.

Associated security issue: Data Loss

- Attackers could recover the deleted data from database or from Camera.

Associated security issue: Data Scavenging

- Attackers could get information from sniffing and spoofing, and could give different information (i.e. images and metadata). Even if we use virtual network for transmission, attackers could listen and access to the images and metadata.

Associated security issue: Transmission and Network Security

- Malicious code can be injected in web application can modify or delete data. Various possible attacks are possible in the Internet can also crash application.

Associated security issue: Application Security

- Virtual Machine created in VMWare could contain malicious code and changes files that affect server and web application or even shutdown the Virtual Machine. This could cause unavailability of the show case.

Associated security issue: Virtual Machine Creation Security

- Web application and server are deployed in Virtual Machine of VMWare. The Virtual Machine could be accessed by a user of other Virtual Machine in the same platform. It means the Virtual Machine is not isolated and all critical information is accessed by the unauthorized user.

Associated security issue: Virtual Machine Isolation Security

- Client's request, server's response and commands to save data in database are executed in Virtual Machine environment. If these communication messages across Virtual Machine environment are interrupted then right operations are also interrupted.

Associated security issue: Virtual Machine Execution Security

- If a Virtual Machine in VMWare is live migrated then attackers could use the exposed state files in the Internet and transfer the Virtual Machine in untrusted host, mishandle client's request and change data.

Associated security issue: Virtual Machine Migration Security

- If Interfaces for requests sent by client and responses sent by server are not standard and does not encapsulate information then critical information is exposed to breaches available in internet.

Associated security issue: Interfaces and APIs Security

- VMWare system administrator has full control. If the administrator change the data, it results in wrong search information and if it deletes the Virtual Machine the whole application is removed.

Associated security issue: Insiders Security

- Transferring data across territorial boundaries should satisfy the laws like "data protection" law. Thus, transferring images and corresponding metadata or videos should also consider laws.

Associated security issue: User-Centric Security

- Since the shared components in the Cloud are not designed for sharing resources. Unauthorized users can access the critical information. If images and corresponding metadata are accessed by unauthorized users then it can be leaked to competitor organization.

Associated security issue: Shared Resources Security

8.3 Selection of Security Issues to be evaluated

Show case has many security issues, among them we select three issues to be evaluated. They are:

- Data Storage Security
- Application Security
- Interfaces and API Security

8.4 SDL Tool selection

We have many security issues in our show case. Selected security issues to be evaluated are Data Storage Security, Application Security, and Interfaces and API Security. In taxonomy, these issues are mapped to SDL standard. From this mapping, we know that these security issues are addressed by SDL. Thus, we choose SDL Tool for evaluation, This selection process is shown in Figure 8.2.

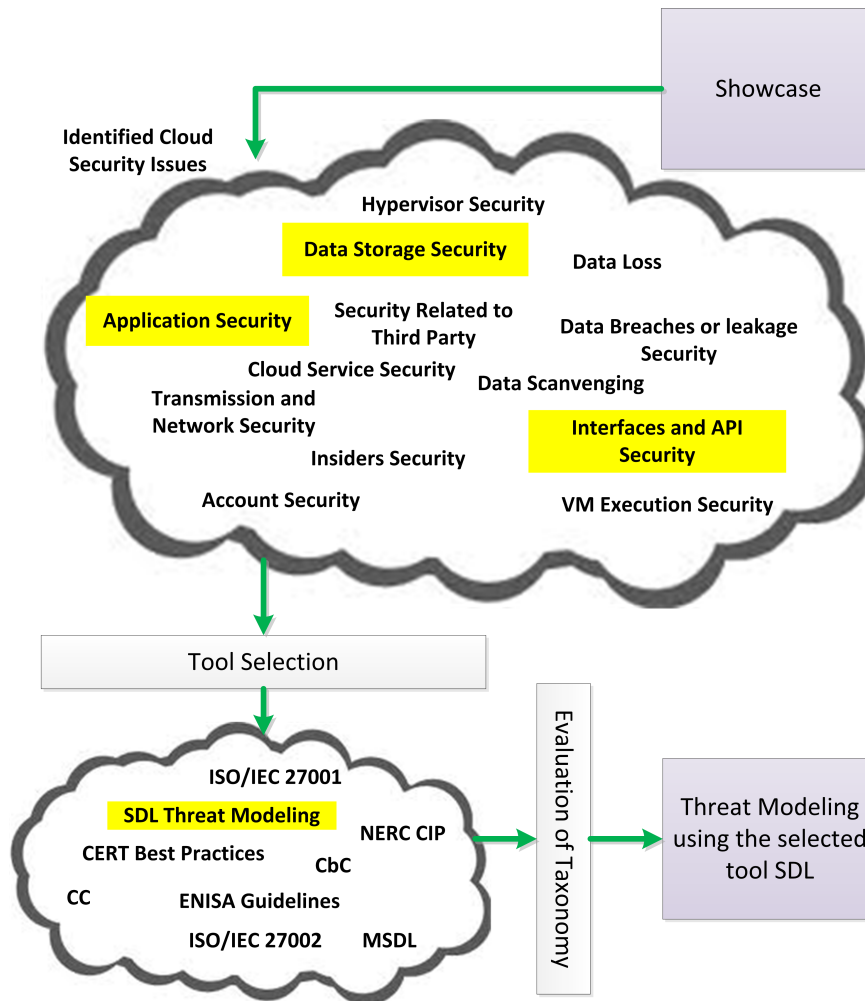


Figure 8.2: Process of tool selection

SDL Tools are great for developing right secure software and these tools sets are used for different purposes in different phases of software development lifecycle. SDL Threat Modeling Tool uses STRIDE and DREAD principle and helps to identify potential threats and mitigation(s) in design phase which is relatively easy, cost-effective to resolve and reduce the total cost of development.

8.5 Threat Modeling

A possible danger that might exploit a vulnerability to breach security and cause possible harm is called a threat. A threat can be either intentional or accidental. Intelligent threat from an individual cracker or a criminal organization is an intentional threat, and the possibility of computer malfunctioning or possibility of natural disaster (e.g., an earthquake, a fire, a tornado) is an example of unintentional or accidental threat.

Threat Modeling is defining a threat model by defining a set of possible attacks to consider while developing a software or a system. Many threat models are defined for a particular software, and these threats and their mitigation(s) are to be considered while developing a secure software. Thus, Threat Modeling is a systematic way to find design-level security and privacy weaknesses in a system. It guides designers or architects to determine correct mitigation(s) and helps to reduce security issues to a system and its data.

Approaches to Threat Modeling

There are different approaches of threat modeling. It depends on what we consider on modeling a threat e.g., software, assets or both. These approaches are as follows:

- *Software-centric*: Software-centric threat modeling is also known as system-centric, design-centric or architecture-centric threat modeling. This approach starts in the design phase of a system. Software architecture diagrams can be illustrated in data-flow diagrams (DFD), use case diagrams or component diagrams. There are different components in threat model and threats to each component are displayed and specific security controls are identified that mitigate the threats considering secure coding standards. An example of software-centric approach is Microsoft's SDL threat modeling tool. We are considering this tool for threat modeling of our show case. We will discuss it in detail in the next sections.
- *Asset-centric*: Identify the assets of an organization or a system. For instance, collection of sensitive personal information, data processed by a system. Assets are classified based on the data sensitivity and their intrinsic value to a potential attacker in order to prioritize risk levels. Attacks trees, attacks graphs are generated in this approach and help to identify multi-step attacks and paths an attackers can reach to asset and prioritize them based on risk levels.
- *Attacker-centric*: In this approach threat modeling requires profiling an attacker's characteristics, skill-set and their motivation to exploit vulnerabilities. These profiles are used to understand attackers, types of exploits and implement mitigation strategy. This approach also uses tree diagrams. Key elements include goals of an attacker, the various considerations related to the system upon which the attack could be perpetrated, along with its software and assets, how the attack could be carried out, and finally, a means to detect or mitigate such an attack.

8.6 SDL Threat Modeling Tool

SDL Threat Modeling Tool allows to identify and mitigate potential security issues in early phase of software development. Therefore, it reduces the cost of development. It is designed for software developers to make development easier by providing guidance on creating and analyzing threat models. It enables software developers or software architects perform the following actions before starting implementation:

- Communicate security designs of their systems
- Analyze designs for potential security issues using a proven methodology
- Suggest and manage mitigation(s) for security issues

Principle of SDL Threat Modeling is STRIDE and DREAD. It generates potential threats based on STRIDE, calculate risk level based on DREAD. So, we can prioritize these threats based on risk level and define mitigation(s) for each threat. These Principles are described in detail in the following sections.

STRIDE

STRIDE categorizes threats in six categories.

- *Spoofing*: A spoofing attack is a situation where a person or a program successfully masquerades as another by falsifying data. Many of the TCP/IP suite do not provide mechanisms for authenticating source and destination of message. They are vulnerable to spoofing attacks like IP Spoofing, ARP Spoofing and may use it for man in the middle attacks. For example, identity spoofing (illegally accessing and then using another user's authentication information, such as username and password).
- *Tempering*: Malicious modification of data is called data tempering. For example, unauthorized changes made to persistent data (like data in database), alteration of data flow between two computers in the Internet.
- *Repudiation*: Repudiation attack happens when a system does not adopt controls to properly track or log users' actions. This allows malicious manipulation or forging users's identification for new actions. So, it changes the authorizing information of actions executed by a malicious user and log wrong information in log files. For example, a user performs an illegal operation in a system but it cannot be traced because there is wrong information (data) in log file. Nonrepudiation is the ability of a system to counter repudiation threats. For example, to buy a item a user has to do signature in the receipt of the item. So, vendor can use the signed receipt as evidence that the user has received the item.
- *Information disclosure*: Information disclosure is about exposing information to the individuals who do not have access to it. For example, the ability of a user to read file which is not granted to access it.

- *Denial of Service*: Denial of service (DoS) attacks deny service to valid users. For example, by making a Web server temporarily unavailable or unusable. Protection against certain types of DoS threats must be done to improve system availability and reliability.
- *Elevation of privilege*: An unprivileged user gains privileged access and has access to compromise or destroy the entire system. An attacker can effectively penetrate all system defense and become itself a part of trusted system. This dangerous situation could be created in Elevation of privilege threat.

While using STRIDE only the first letter of these threats are used, **S** for Spoofing, **T** for Tampering, **R** for Repudiation, **I** for Information disclosure, **D** for Denial of Service, and **E** for Elevation of privilege.

DREAD

DREAD is a risk rating model in threat modeling. Formula used to compute a risk value by computing the average value of all five categories is as follows:

Risk DREAD = (DAMAGE POTENTIAL + REPRODUCIBILITY + EXPLOITABILITY + AFFECTED USERS + DISCOVERABILITY) / 5

- *Damage Potential*: How much damage will be caused occurring the threat exploit? i.e. how severe is the attack?
0 = Nothing.
5 = Individual user data is compromised or affected.
10 = Complete system or data destruction.
- *Reproducibility*: How easy is to reproduce the threat exploit?
0 = Very hard or impossible, even for administrators of the application.
5 = One or two steps required, may need to be an authorized user.
10 = Just a web browser and the address bar is sufficient, without authentication.
- *Exploitability*: How hard is it to work out how to attack?
0 = Advanced programming and networking knowledge, with custom or advanced attack tools.
5 = Malware exists on the Internet, or an exploit is easily performed, using available attack tools.
10 = Just a web browser
- *Affected Users*: How many users are affected?
0 = None
5 = Some users, but not all
10 = All users
- *Discoverability*: How easy is it to discover this threat?
0 = Very hard to impossible; requires source code or administrative access.
5 = Can figure it out by guessing or by monitoring network traces.

9 = Details of faults like this are already in the public domain and can be easily discovered using a search engine.

10 = The information is visible in the web browser address bar or in a form.

While using DREAD we use only the first letter of these categories, **D** for Damage Potential, **R** for Reproducibility, **E** for Exploitability, **A** for Affected Users and **D** for Discoverability.

Process of SDL Threat Modeling

SDL Threat Modeling has various steps. Main four steps of the process is shown in Figure 8.3.

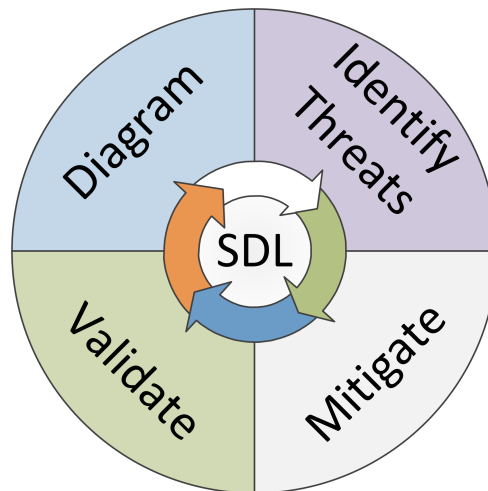


Figure 8.3: SDL Threat Modeling Process.

1. **Diagram:**

Diagram Elements:

Diagram elements used in data flow diagram are:

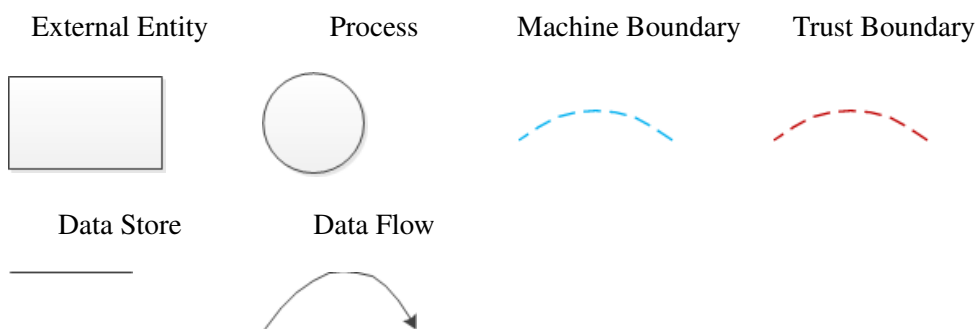


Diagram Layers:

- **Context Diagram:** Very high level diagram of a system. It contains entire components of the system.
 - **Level 1 Diagram:** High level diagram showing the single feature or scenario.
 - **Level 2 Diagram:** Low level diagram containing detailed sub-components of features.
 - **Level 3 Diagram:** It is more detailed diagram. More layers are rarely needed except in huge projects. In this detailed diagram more trust boundaries are drawn.
2. **Identify Threats:** Tool generates potential threats for each elements of the diagram. It uses STRIDE to step through the diagram elements. For each type of threats, we want different property and mitigation(s) to maintain the property. For each of the STRIDE threat, wanted property is in Table 8.1.
 3. **Mitigate:** Mitigation is the point of threat modeling. Mitigation is about addressing a problem and designing a secure software. We can apply standard mitigation(s) or invent new mitigation(s) for each of the threats. After addressing or defining mitigation(s) for all the potential threats in a system or a software, it is redesigned to eliminate the threats. Some standard mitigation(s) for each of the threats is shown in Table 8.1.
 4. **Validate:** In this phase of the process, we validate the whole threat modeling such that diagram match with the final code, all threats are mitigated? Additionally, validate the quality of threats and their mitigation(s).

Threat	Property we want	Standard mitigation(s)
Spoofing	Authentication	Cookie authentication, IPsec, Windows Authentication (NTLM), PKI System such as SSL/TLS and certificates, digital signatures, message authenticate codes
Tampering	Integrity	Windows Mandatory Integrity Control, Access Control Lists-ACLs, Digital Signatures
Repudiation	Non-Repudiation	Secure logging and auditing, Digital signatures, Secure time stamps
Information Disclosure	Confidentiality	Encryption, ACLs
Denial of Service	Availability	High availability designs, Quotas, Filtering
Elevation of Privilege	Authorization	ACLs, Group or role membership, Privilege ownership, Permission, Input validation

Table 8.1: STRIDE threats generated based on the Entity type.

Apply in Show Case

We develop a Data Flow Diagram -DFD of our show case. The DFD contains all components Client, Server, Web Application and Elastic Database.

Figure 8.4 is a DFD of our show representing data flow between different components, different kinds of boundaries, processes and user. Client is installed in a Gateway. All CCTV cameras are connected to the Gateway and all videos from CCTV cameras are sent to it. In Figure 8.4, arrows from CCTV cameras to Gateway shows data flow or videos from cameras to the Gateway. Blue dotted lines are machine boundaries and red dotted lines are trust boundaries. Each CCTV camera is a different machine, so there is a machine boundary between the cameras. Gateway is a separate machine, cameras and Gateway may not be in same place. So there are machine boundary and trust boundary between CCTV cameras and Gateway. Fake videos could be sent to Gateway, so should verify that incoming videos are the authenticated videos. Images and corresponding metadata are extracted in Gateway. Image and corresponding metadata is sent to Server as a Request. It is represented by Request arrow from Gateway to next process (Get images and metadata from request) which is in Server. Server is deployed in the Cloud, so this data transmission is in the Internet where both machine boundary and trust boundary exist. Data from the Request is saved to Elastic Database in the Cloud. Save arrow represent this operation in DFD. After saving it in the Database, Response is sent to previous process and the process sends it to the Gateway. User uses search form of the Web Application to search images that are saved in the Cloud. User types available metadata and submits the search form. This operation is represented by arrow from user to browser. The submitted data is sent to the Cloud to perform search operation in Elastic Database. Flow of metadata from browse to the Cloud is represented by arrow from browser to Web Application Server. This data flow is in the Internet which requires both trust boundary and machine boundary represented by red and blue dotted lines. Search Query is sent to the Database and result/data after fetching is sent back to Web Application. These operations are represented by arrows from Web Application to Database and from the Database to the Web Application. Result from this search operation displays in browser and user can view images and corresponding information.

Different threats affect each element type. Threats according to element type is shown in Table 8.2. Possible threats for External Entity are Spoofing and Repudiation and all threats Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service, Elevation of privilege are possible for Process. Similarly, possible threats for Data Store and Data Flow are marked in the Table.

Element	S	T	R	I	D	E
External Entity	✓		✓			
Process	✓	✓	✓	✓	✓	✓
Data Store		✓	✓	✓	✓	
Data Flow		✓		✓	✓	

Table 8.2: STRIDE threats generated based on the Entity type.

From table 8.2, we know the possible threats for the elements in DFD of our show case.

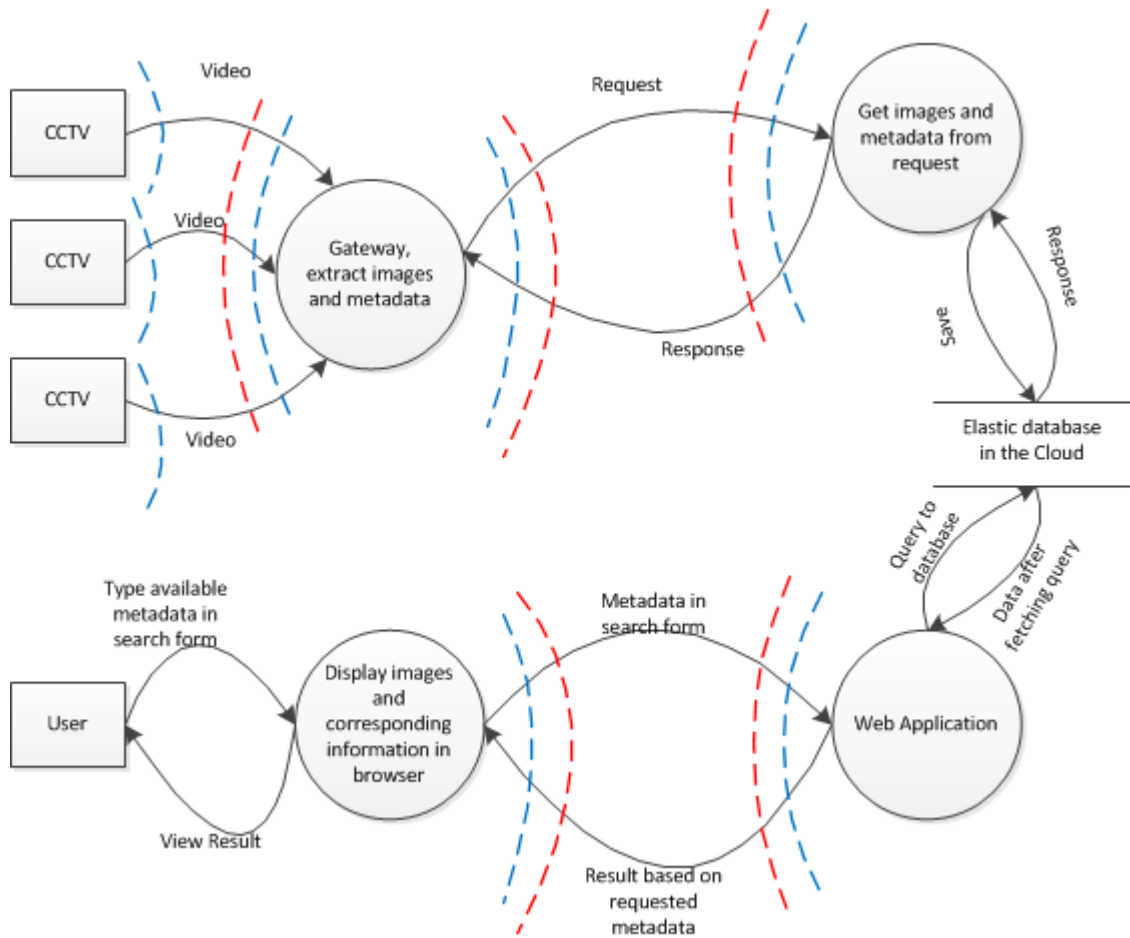


Figure 8.4: Data flow diagram of show case described in Chapter 5.

Figure 8.5 shows the possible threats based on the element type in the DFD of our show case. CCTV and user are external entities, possible threats for CCTV and user are Spoofing and Repudiation (SR). Possible threats for process (Gateway extract images and metadata) are Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service and Elevation of privilege (STRIDE). Similarly, STRIDE is generated for other processes (Get images and metadata from request, Web Application, Display images and corresponding information in browser). Elastic database in the Cloud is a Data Store element in the DFD. Possible threats in Elastic database are Tampering, Repudiation, Information disclosure and Denial of Service (TRID). Data flow from CCTV to Gateway is transferring video from CCTV to Gateway and possible threats are Tampering, Information disclosure and Denial of Service (TID). Similarly, for data flow from Gateway to Server, Server to Elastic Database and back to Gateway has TID possible threats. Not only from Client/Server side but also in User to Web Application data flow between elements has TID possible threats.

In SDL Threat Modeling Tool, all of these possible threats are generated for all elements of

DFD and we define impacts and solutions for each of the potential threats. This helps us to know the potential threats of our show case, impacts of the threats and their solutions. Figure 8.5 is only a graphical representation of threats, defining their impacts and solutions or mitigation(s) is the main task of this approach. In Analyze Model section, this tool lists all of the elements and corresponding potential threats where impacts and mitigation(s) are to be defined. We have generated a long report on this, a part of this report is listed in Appendix.

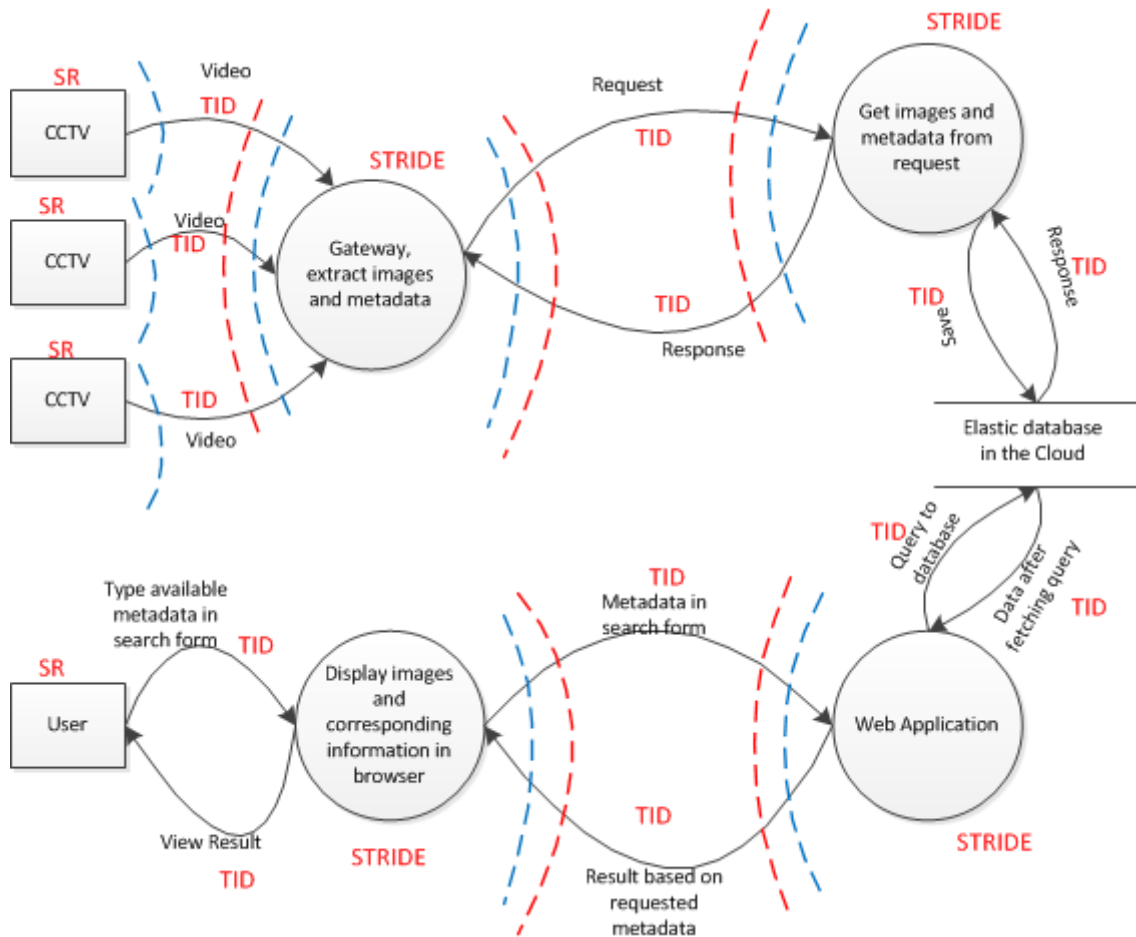


Figure 8.5: Data flow diagram of show case described in Chapter 5 with potential threats.

Entitiy Affected	ID	Name	D	R	E	A	D	Rating
Elastic database	103	Attacker able to modify images and corresponding metadata in database.	10	5	0	10	0	5
Elastic database	104	Changes made in database cannot be traced.	5	5	0	5	5	4
Web application	119	Attacker access hidden fields' sensitive information and use it for attack.	5	10	0	5	9	5.8
Web application	120	Web Application is slow or shutdown such that it cannot establish further (more) connections.	10	5	10	10	0	7
Web application	121	Admin user of web server access the database from shared memory.	10	0	0	10	9	5.8

Table 8.3: Calculation of risk level based on DREAD formula.

We have also calculated risk level of some of the threats generated in the show case. Risk level of the threats helps us to priorities them and define the mitigation(s). Risk level for some of the threats in our show case are shown in Table 8.3. We have calculated risk level of the threats of entities Elastic database (Data Store) and Web Application (Process). ID is the threat number while generating the potential threats. We describe these threats briefly and then calculate risk values for Damage Potential, Reproducibility, Exploitability, Affected Users and Discoverability (DREAD) and rate them. Now, we show how risk level of these threats are calculated according to DREAD principle.

Threat 103 is a Elastic database threat. If an attacker is able to modify images and corresponding metadata in elastic database then this threat occurs. Affect of the threat is calculated as follows:

Damage Potential = Complete system or data destruction = 10

Reproducibility = One or two steps required, may need to be an authorized user = 5

Exploitability = Advanced programming and networking knowledge, with custom or advanced attack tools = 0

Affected Users = All users = 10

Discoverability = Very hard to impossible; requires source code or administrative access = 0

Risk Level (Rating) = (Damage Potential + Reproducibility + Exploitability + Affected User + Discoverability)/5 = (10 + 5 + 0 + 10 + 0)/5 = 5

Threat 119 is a Web Application threat. If this threat occurs then an attacker is able to access sensitive information of hidden fields and uses this information for attack. Affect of this threat is calculated as follows:

Damage Potential = Individual user data is compromised or affected = 5

Reproducibility = Just a web browser and the address bar is sufficient without authentication = 10

Exploitability = Advanced programming and networking knowledge, with custom or advanced attack tools = 0

Affected Users = Some users but not all = 5

Discoverability = Details of faults like this are already in the public domain and can be easily discovered using a search engine = 9

Risk Level (Rating) = (Damage Potential + Reproducibility + Exploitability + Affected User + Discoverability)/5 = (5 + 10 + 0 + 5 + 9)/5 = 5.8

Threat 120 occurs in Web Application. If Web Application is slow or shutdowns and cannot establish further connections then this threat occurs. Affect of the occurrence of this threat is calculated as follows:

Damage Potential = Complete system or data destruction = 10

Reproducibility = One or two steps required, may need to be an authorized user = 5

Exploitability = Just a web browser = 10

Affected Users = All users = 10

Discoverability = Very hard to impossible; requires source code or administrative access = 0

Risk Level (Rating) = (Damage Potential + Reproducibility + Exploitability + Affected User + Discoverability)/5 = (10 + 5 + 10 + 10 + 0)/5 = 7

Similarly, Ratings of threats 104 and 121 are calculated. According to the values in Rating column, we can prioritize them as 120, 119, 121, 103, 104. Threat 120 has high risk level, so it has highest priority. From this Table we can calculate their priority and apply in system development.

8.7 Contribution of SDL Threat Modeling to address Security Issues

We want to develop a secure software that has built in security while developing it. For this purpose we choose SDL/MSDL, which is appropriate for every phase of software development lifecycle. We choose SDL Threat Modeling Tool to experiment our show case. It is a software centric Threat Modeling Tool. It generates potential threats to each of the components of the Data-Flow-Diagram. Mitigation(s) for each of the potential threats suggested by security experts, and implementing them while developing software ensures security in a system or a software. We evaluate mapping of security issues and secure software development means in Chapter 7.

- *Data Storage*: Data-Store component of SDL Threat Modeling Tool corresponds to database. Regarding the Data-Store component, it generates TID-Tampering, Information Disclosure and Denial of Service threats. Mitigation(s) applied to the potential threats generated to Data-Store component help us to secure operations of database and content in database. In our show case, it secures operations relating to the elastic database in the cloud and secure database.
- *Application Security*: In SDL Threat Modeling, an application or a software is presented in a Data-Flow-Diagram. Thus, a software is presented with all the processes, users, databases and data flows (how information flows between the elements). Cyber attacks can not only crash systems but also cause physical damage. Applying standard mitigation(s) for SQL Injections, Cross-Site-Scripting, Buffer overflow, Man-in-the-middle attacks etc.

help us to develop a secure software. All these solutions to the attacks possible in the Internet are suggested by security experts in design phase and are implemented in development phase. Not only mitigation(s), bugs in a software can also be traced in a file via bug tracking system. All of these steps in SDL Threat Modeling Tool help us to develop a secure software and support for application security. In the show case, we use the web search application deployed in cloud environment applying the suggested mitigation(s), secure the application and avoid the attacks.

- *Interfaces and API Security*: Interfaces and API of a system or a software are needed to be standard and proactive. Strong authentication and access control list implemented in a software help us to make API more secure. Prevention of unauthorized access, changes, process or configuration secure the systems. These systems might be the components of Critical Infrastructure (e.g., control systems). Understanding the dependency of components and chain associated with API and applying mitigation(s) of potential threats help to secure Interfaces and API. Standard interfaces in show case, strong authentication accessing the Cloud environment is supported by providing the possible threats and mitigation(s).

Summary and Conclusion

9.1 Summary

Secure software development standards, guidelines and tools are applicable in different context. Our study of evaluation of these secure software development means is in the context of Critical Infrastructure and the Cloud. Thus, we have evaluated these secure software development means in the context of Critical Infrastructure and the Cloud. We have identified a list of security issues in the context of Critical Infrastructure and the Cloud from a literature review and a show case analysis. For each of the issues, we have investigated on the applicability of popular secure software development means and map the issue and the standards, guidelines or tools that address the particular issue. We have applied SDL Threat Modeling Tool in our show case to evaluate the mappings of security issues and secure software development means. A multidimensional taxonomy of the secure software development means and the issues is defined based on the mapping. We have categorized these means to standards, guidelines and tools, and further sub-categorized them into software development and security.

9.2 Conclusion

Critical Infrastructure is rapidly integrating in the Cloud to benefit from characteristics of the Cloud. This integration has security issues, we have identified the security issues and investigated on the applicability of different secure software development means that address these security issues. Each security issue is mapped to supporting standards, guidelines and tools depending on which standards, guidelines and tools help to mitigate the security issue. We present these mapping in the form of multidimensional taxonomy. Mapping of the security issues and secure software development means in the taxonomy is evaluated by applying SDL Threat Modeling Tool in our show case. We have bounded our study on some popular secure software development means.

Our output helps Critical Infrastructure and Cloud providers or stakeholders to select the right means to build a secure software for the given context.

9.3 Future Work

We are extending the multidimensional taxonomy in Figure 7.2, and designing new mechanisms for developing secure software in the Cloud using the multidimensional taxonomy. Main goals of this work are:

- Develop a Tool Chain and methods to support the semi-automatic selection of standards and tools for secure software development of secure “High Assurance” cloud applications.
- Develop a code generation mechanism like in the above point based on the considered security aspects, code fragment for “High Assurance” cloud applications.

Appendix

A.1 Standards

At the beginning of research, we have tried to investigate on more standards and guidelines. Later on, we have bounded our research on popular secure software standards and guidelines. Some of the standards are as follows:

ISA Standards

International Society of Automation (ISA) was established in 1945 and is a non-profit society of engineers, technicians, businesspeople, educators and students who are interested in industrial automation. ISA is a professional organization for setting standards and educating industry professionals in automation. Some examples of ISA standards are ISA99 (a cyber security standard for control systems), ISA95 (standard for developing an automated interface between enterprise and control systems). ISA has also formed ISA Security Compliance Institute to promote cyber-secure products and practices for industrial automation suppliers and operational sites.

- *ISA99*: ISA99 [77] is a cyber security standard for control systems produced by ISA. Cyber security standards provide security techniques to minimize cyber security attacks. Cyber security depends on Industrial control system which is important to the Critical Infrastructure. ISA99 committee addresses on industrial and automation systems focusing on the improvement of Confidentiality, Integrity and Availability of control systems and provides criteria for secure control system.

ANSI Standards

American National Standards Institute - ANSI is a private non-profit organization in United States. It coordinates U.S. standards with international standards so that American products

can be used worldwide. ANSI not only accredits standards developed from other standard organizations, government agencies, companies and many others but also accredits organizations carrying out product or personnel certifications with requirements defined in international standards. ANSI is the official U.S. representative to the two major international standards (ISO and IEC) via the U.S. National Committee (USNC) and participates in almost activities of ISO and IEC. Many U.S. standards are taken to ISO and IEC through ANSI and USNC.

ANSI/ISA Standards

Many ISA standards have been recognized by ANSI and also have been adopted as IEC standards. For instance, i) *ANSI/ISA-50.02-Fieldbus Standard for Use in Industrial Control Systems* is a product of *ISA-50- Signal Compatibility of Electrical Instruments committee*, and ii) *ANSI/ISA-88.00.01 Models and Terminology* is a batch processing standard, *ANSI/ISA-88.00.02 Data Structures and Guidelines for Languages* and *ANSI/ISA-88.00.03 General and Site Recipe Models and Representation* are products of the *ISA-88 Batch Control committee*.

Cloud Security Alliances (CSA)

CSA is an not-for-profit oraganization established in 2008. CSA has two main objectives: One is to promote the use of best practices for providing security assurance and other is to provide education of cloud computing that helps to secure computing. CSA describes a set of best security practices in “Security Guidance for Critical Areas of Focus in cloud computing” [78].

A.2 Applying SDL Threat Modeling Tool in Show Case

We have applied SDL Threat Modeling Tool in our show case. Data Flow Diagram of the show case is used in the tool and potential threats and mitigation(s) are defined for each of the entities. Report is generated considering the Data Flow Diagram, all the defined threats and their mitigation(s). If there are certificates defined for any threats and claim that the threat type is not possible then it is also generated in the report. Threat Model Information is about where and by whom the threats are modeled. It is also considered in the report. Here, we are presenting structure of the report.

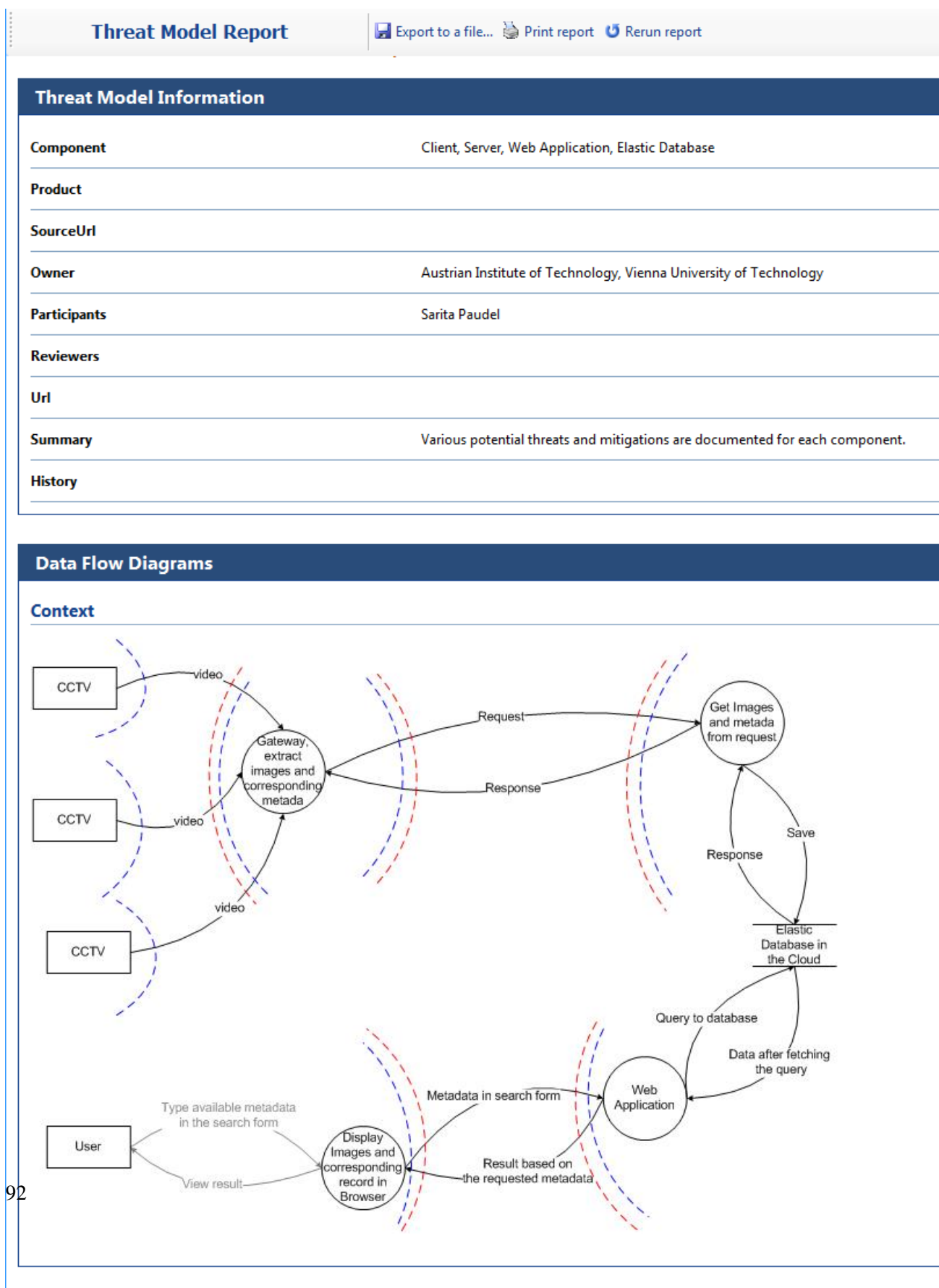


Figure A.1: Threat Model Report: Threat Model Information and Data Flow Diagram.

External Interactors

Threats against CCTV

Spoofing (Threat #38)

Threat: If spoofing video is sent from unauthenticated CCTV. Expected information from images and metadata are lost.

Mitigation: CCTV LiveID authentication and sending digitally signed packets to gateway.

Repudiation (Threat #39)

Threat: Somebody else could send video in using authentication of CCTV wrong provide wrong log information.

Mitigation: Non-repudiation by strong authentication while connecting to CCTV, secure logging and auditing, digital signature while sending video to gateway.

Threats against User

Spoofing (Threat #140)

Threat: Spoofing user can perform the malicious actions and provide malicious code in search form.

Mitigation: Cookie authentication, strong authentication with digital signature is used to avoid spoofing.

Repudiation (Threat #141)

Threat: User can perform malicious actions that are not traced in user log files.

Mitigation: User should be trusted, user authentication, logging and auditing of user helps avoid repudiation.

Processes

Threats against Display Images and corresponding record in Browser

Spoofing (Threat #128)

Threat: Information may be displayed sent by spoofed server.

Mitigation: Establish connection with LiveId Authentication via SSL certificated transmission.

Tampering (Threat #129)

Threat: Wrong information displayed in browser. Attack in the internet (e.g. man-in-the-middle) changes original information and send it to the user.

Mitigation: Use authentication codes or digital signatures while sending the search form parameters to the from user to the web server.

Repudiation (Threat #130)

Threat: User performs malicious actions without leaving its identity and malicious actions log.

Mitigation: Maintain Action Control List and perform secure logging and auditing of users.

Information Disclosure (Threat #131)

Threat: Sensitive data of previous user is displayed to another another (may be from history)

Mitigation: Implement session management correctly, add cache control and make sure that forms and every sensitive variable are submitted only through POST requests and not GET.

Denial of Service (Threat #132)

Threat: User browsing the search web application face internet browsing.

Figure A.2: Threat Model Report: Threats and Mitigation(s).

Mitigation: Assign users to groups and roles with different permissions, and also perform the user input validation.

Threats against Gateway, extract images and corresponding metadata

Spoofing (Threat #46)

Threat: Spoofing of gateway identity and establishing connections to CCTVs lost all videos generated and third party access the videos.

Mitigation: Windows Authentication(e.g. NT Lan Manager-NTLM(authenticated window have special functionality)), cookie authentication, LiveID authentication should implemented as solutions.

Tampering (Threat #47)

Threat: Videos are integrated in a list according to incoming order in gateway. This list could be changed and lost the videos in the list before retrieving images and corresponding metadata. If shared memory in

Mitigation: Message authentication code in videos or digital signatures in the video packets helps in avoid tampering.

Repudiation (Threat #48)

Threat: Unauthorized actions like deleting video could be done in gateway and it cannot be traced in log files.

Mitigation: Nonrepudiation secure time stamps, digital signatures are used and log the actions in log files.

Information Disclosure (Threat #49)

Threat: Unauthorized user in gateway access shared memory which cause information leakage.

Mitigation: Gateway access should be maintained in ACL (Access Control List).

Denial of Service (Threat #50)

Threat: CCTV could not establish connection with gateway and send generated streaming video. As a consequence, videos are lost.

Mitigation: Design gateway with high availability.

Elevation of Privilege (Threat #51)

Threat: User having access to the gateway application configuration could delete incoming streaming videos.

Mitigation: Maintain ACL (Access Control List), assign each user in a group and an individual role, and different privilege to different roles.

Threats against Get Images and metadata from request

Spoofing (Threat #88)

Threat: While extracting images and corresponding metadata before saving in database, spoofed images and corresponding metadata can be injected.

Mitigation: Send digitally signed images and corresponding metadata and check it in while extracting from request.

Tampering (Threat #89)

Threat: If images and corresponding metadata are changed then wrong information will be saved in database.

Mitigation: Check integrity by digital signature or message authentication code.

Repudiation (Threat #90)

Threat: Original data sent in request is changed and not traced before saving.

Figure A.3: Threat Model Report: Threats and Mitigation(s).

Denial of Service (Threat #42)

Threat: If many unnecessary connections are established in the gateway and it cannot establish connection to the authorized CCTV for sending video.

Mitigation: Design high availability of gateway, maintain the quotas of active CCTV to connect to gateway.

Threats against video

Tampering (Threat #43)

Threat: Videos are altered and metadata of video could be changed while sending to the gateway.

Mitigation: Check integrity by using windows mandattory controls or accept the videos in gateway that digitally signed.

Information Disclosure (Threat #44)

Threat: Celebrity video is accessed falls in wrong hands who misuse the information in the video.

Mitigation: Maintatin confidentiality by encryption and by Access Control List (ACL) to access the videos.

Denial of Service (Threat #45)

Threat: If many unnecessary connections are established in the gateway and it cannot establish connection to the authorized CCTV for sending video.

Mitigation: Design high availability of gateway, maintain the quotas of active CCTV to connect to gateway.

Data Stores

Threats against Elastic Database in the Cloud

Tampering (Threat #103)

Threat: If attacker is able to alter data in database, then images and corresponding metadata could be repalced by new ones or make it unreadable.

Mitigation: Claculate checksum for database and tables, last time access, digital signature for access. If the checksum is different then data is altered. Strong authentication and ACL (Access control list) with differ

Repudiation (Threat #104)

Threat: Unknown user makes changes on data and changes are not traced.

Mitigation: Secure logging and auditing of database operations.

Information Disclosure (Threat #105)

Threat: Sensitive data like personal identity, credit card is accessed to normal users. This critical information can be misused.

Mitigation: Database encription.

Denial of Service (Threat #106)

Threat: Information cannot be retrieved from database then services using this database are also not working properly.

Mitigation: Design database with high availability.

Certifications

User Alias	Element Name	Threat Type	Reason for certification	Description
PaudelS	Save	DenialOfService	within a trust boundary	Data is extrected from request validate data using digital signature and then only send d

Figure A.4: Threat Model Report: Threats and Mitigation(s) and Certifications.

Bibliography

- [1] R. Fisher J. Peerenboom and R. Whitfield. Recovering from disruptions of interdependent critical infrastructures. In *CRIS/DRM/IHIT/NSF Workshop on Mitigating the Vulnerability of Critical Infrastructures to Catastrophic Failures*, 2001.
- [2] Microsoft Cybersecurity Homepage. 7 Steps for Critical Infrastructure Protection. <http://www.microsoft.com/security/cybersecurity/>, 2014. [Online; accessed January 2014].
- [3] B. Solomon, D. Ionescu, M. Litoiu, and G. Iszlai. Designing autonomic management systems for cloud computing. In *Computational Cybernetics and Technical Informatics (ICCC-CONTI), 2010 International Joint Conference on*, pages 631–636, 2010.
- [4] Ross Anderson. *Security Engineering: A guide to Building Dependable Distributed Systems*. Wiley Publishing, Inc., Indianapolis, IN 46256, USA, 2008.
- [5] Noopur Davis. Secure software development life cycle processes: A technology scouting report. In *Secure Software Development Life Cycle Processes: A Technology Scouting Report*, 2005. [Online; accessed 15-July-2013].
- [6] Michael Howard and Steve Lipner. *The Security Development Lifecycle*. Microsoft Press, Redmond, WA, USA, 2006.
- [7] Microsoft Secure Development Lifecycle. Simplified implementation of the microsoft sdl. 2010. [Online; accessed 16-July-2013].
- [8] Microsoft Secure Development Lifecycle. Microsoft security development lifecycle tools. [Online; accessed 22-July-2013].
- [9] A. Hall and R. Chapman. Correctness by construction: developing a commercial secure system. *Software, IEEE*, 19(1):18–25, 2002.
- [10] CCRA Management Committee. Common Criteria for Information Technology Security Evaluation. <http://www.commoncriteriaportal.org/cc/>, 2013. [Online; accessed 16-July-2013].
- [11] W. David Snead P.C. Paolo Balboni, Kieran Mccorry. Benefits, risks and recommendations for information security. pages 7–9, 71–77, 2009.

- [12] ENISA Giles Hogben, Marnix Dekker. A guide to monitoring of security service levels in cloud contracts. 2012.
- [13] Carnegie Mellon Software Engineering Institute. Computer emergency response team - cert. [Online; accessed 23-July-2013].
- [14] Andrew Moore Randall Trzeciak Timothy J. Shimeall Lori Flynn George Silowash, Dawn Cappelli. Common sense guide to mitigating insider threats. [Online; accessed 23-July-2013].
- [15] Carnegie Mellon Software Engineering Institute. Octave method. [Online; accessed 23-July-2013].
- [16] International Organization for Standardization and International Electrotechnical Commission. Iso/iec 27001:2005 information technology — security techniques — information security management systems — requirements. 2005. [Online; accessed 23-July-2013].
- [17] ISO/IEC. Iso/iec 27001:2005 information technology — security techniques — information security management systems — requirements. 2005. [Online; accessed 23-July-2013].
- [18] ISO. Iso/iec 27002:2005 information technology — security techniques — code of practice for information security management. 2005. [Online; accessed 23-July-2013].
- [19] Sarita Paudel, Markus Tauber, and Ivona Brandic. Security standards taxonomy for cloud applications in critical infrastructure it. In *Internet Technology and Secured Transactions (ICITST), 2013 8th International Conference for*, pages 645–646, Dec 2013.
- [20] Lynn Fitcher and Rossouw von Solms. Guidelines for secure software development. In *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*, SAICSIT '08, pages 56–65, New York, NY, USA, 2008. ACM.
- [21] National Institute of Standards and Technology. The nist definition of cloud computing. [Online; accessed 16-June-2014].
- [22] Josh Ames. Types of cloud computing: Private, public and hybrid clouds. In *Types of Cloud Computing*, 2012. [Online; accessed 19-December-2013].
- [23] International Business Machines Corporation. Cloud computing service models. [Online; accessed 17-June-2014].
- [24] Microsoft. Office 365. In *Microsoft Office 365*. [Online; accessed 27-December-2013].
- [25] Michael Howard and Steve Lipner. *The Security Development Lifecycle*. Pearson Education, Inc., Addison-Wesley, United States of America, 2011.

- [26] Open Web Application Security Project. Open web application security project. [Online; accessed 17-June-2014].
- [27] J.Y. Liang and P.X. Wu. Challenges of cloud computing evaluation. *Applied Mechanics and Materials*, 198-199:396–401, 2012.
- [28] Kui Ren, Cong Wang, and Qian Wang. Security challenges for the public cloud. *Internet Computing, IEEE*, 16(1):69–73, 2012.
- [29] M.A.C. Dekker. Critical cloud computing:ciip perspective on cloud computing. 2013. [Online; accessed 19-July-2013].
- [30] ImadM. Abbadi. Toward trustworthy clouds’ internet scale critical infrastructure. In Feng Bao and Jian Weng, editors, *Information Security Practice and Experience*, volume 6672 of *Lecture Notes in Computer Science*, pages 71–82. Springer Berlin Heidelberg, 2011.
- [31] Madjid Merabti Younis A.Younis and Kashif Kifayat. Secure cloud computing for critical infrastructure: A survey. Technical report, Liverpool John Moores University, United Kingdom, 2013.
- [32] Nuno Santos, Rodrigo Rodrigues, Krishna P. Gummadi, and Stefan Saroiu. Policy-sealed data: A new abstraction for building trusted cloud services. In *USENIX Security*, 2012.
- [33] B.R. Kandukuri, V.R. Paturi, and A. Rakshit. Cloud security issues. In *Services Computing, 2009. SCC '09. IEEE International Conference on*, pages 517–520, 2009.
- [34] Rajkumar Buyya, Rodrigo N. Calheiros, and Xiaorong Li. Autonomic cloud computing: Open challenges and architectural elements, 2012.
- [35] Coimbatore Chandrasekaran, William R Simpson, and Ryan R Wagner. High assurance challenges for cloud based computing. In *Proceedings of the World Congress on Engineering and Computer Science 2011 , October 19-21, 2011, San Francisco, USA*, volume Vol I WCECS 2011, 2011.
- [36] S.R. Lenkala, S. Shetty, and Kaiqi Xiong. Security risk assessment of cloud carrier. In *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, pages 442–449, 2013.
- [37] Jia Yu and Rajkumar Buyya. A taxonomy of workflow management systems for grid computing. Technical report, 2005.
- [38] Robert Dukaric and Matjaz B. Juric. Towards a unified taxonomy and architecture of cloud frameworks. *Future Generation Computer Systems*, 29(5):1196 – 1210, 2013. Special section: Hybrid Cloud Computing.
- [39] Reijo M. Savola. Towards a taxonomy for information security metrics. In *Proceedings of the 2007 ACM workshop on Quality of protection, QoP '07*, pages 28–30, New York, NY, USA, 2007. ACM.

- [40] M. Razzazi, A. Tahouri, and K. Fayazbakhsh. Evaluation process management software for security evaluation. In *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on*, pages 1–4, 2008.
- [41] P.A. Karger, S. McIntosh, E.R. Palmer, D. Toll, and S. Weber. Lessons learned: Building the caernarvon high-assurance operating system. *Security Privacy, IEEE*, 9(1):22–30, 2011.
- [42] Martin Johns, Christian Beyerlein, Rosemaria Giesecke, and Joachim Posegga. Secure code generation for web applications. In Fabio Massacci, Dan Wallach, and Nicola Zannone, editors, *Engineering Secure Software and Systems*, volume 5965 of *Lecture Notes in Computer Science*, pages 96–113. Springer Berlin Heidelberg, 2010.
- [43] Checkmarx. We secure your code. 2014. [Online; accessed 29-January-2014].
- [44] Crunch Base. Secure code generator. 2014. [Online; accessed 29-January-2014].
- [45] P. Saripalli and B. Walters. Quirc: A quantitative impact and risk assessment framework for cloud security. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pages 280–288, 2010.
- [46] Trusted Computing Group. Trusted computing group. 2010. [Online; accessed 11-Oct-2013].
- [47] Carlos Cardenas. The four keys of cloud security. 2013. [Online; accessed 14-June-2013].
- [48] David Molnar and Stuart Schechter. Self hosting vs. cloud hosting: Accounting for the security impact of hosting in the cloud.
- [49] Stephen MacDonell, Martin Shepperd, Barbara Kitchenham, and Emilia Mendes. How reliable are systematic reviews in empirical software engineering? *IEEE Transactions on Software Engineering*, 36(5):676–687, 2010.
- [50] Robert K. Yin. *Case Study Research: Design and Methods*. Applied social research methods series. Sage Publications, Beverly Hills, CA, 1984.
- [51] S.S. Qureshi, T. Ahmad, K. Rafique, and Shuja ul islam. Mobile cloud computing as future for mobile applications - implementation methods and challenging issues. In *Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on*, pages 467–471, 2011.
- [52] Bilal Al Baalbaki, Youssif Al-Nashif, Salim Hariri, and Douglas Kelly. Autonomic critical infrastructure protection (acip) system. In *Computer Systems and Applications (AICCSA), 2013 ACS International Conference on*, pages 1–4, 2013.
- [53] H. Tianfield. Security issues in cloud computing. In *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, 2012.

- [54] Mutum Zico Meetei and Anita Goel. Security issues in cloud computing. In *Biomedical Engineering and Informatics (BMEI), 2012 5th International Conference on*, pages 1321–1325, 2012.
- [55] K. Popovic and Z. Hocenski. Cloud computing security issues and challenges. In *MIPRO, 2010 Proceedings of the 33rd International Convention*, pages 344–349, 2010.
- [56] S. Sakr and A. Liu. Sla-based and consumer-centric dynamic provisioning for cloud databases. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 360–367, 2012.
- [57] CLOUD SECURITY ALLIANCE. The notorious nine: Cloud computing top threats in 2013. 2013. [Online; accessed 10-Oct-2013].
- [58] Keiko Hashizume, DavidG Rosado, Eduardo Fernández-Medina, and EduardoB Fernandez. An analysis of security issues for cloud computing. *Journal of Internet Services and Applications*, 4(1):1–13, 2013.
- [59] OWASP. Open web application security project. 2013. [Online; accessed 28-Nov-2013].
- [60] Certification Europe. Information security management systems. In *ISO-27001:2005*. [Online; accessed 27-November-2013].
- [61] ISO 27001 Security. Iso 27002 compliance software. In *ISO 27002:2005*. [Online; accessed 27-November-2013].
- [62] North American Electric Reliability Corporation. Nerc cip standards. [Online; accessed 20-August-2013].
- [63] Daniel Mellado, Eduardo Fernández-Medina, and Mario Piattini. A common criteria based security requirements engineering process for the development of secure information systems. *Computer Standards Interfaces*, 29(2):244 – 253, 2007.
- [64] Josh Ames. Secure development lifecycle certificates. In *Secure Development Lifecycle*, July 2011. [Online; accessed 19-December-2013].
- [65] Josh Ames. Cert. In *Computer Emergency Response Team*. [Online; accessed 20-December-2013].
- [66] Josh Ames. Enisa. In *European Network and Information Security Agency*, November 25 2011. [Online; accessed 19-December-2013].
- [67] Air Magnet. Report iso 27001. 2013. [Online; accessed 09-January-2014].
- [68] ISO 27001. Iso 27001 controls and objectives. [Online; accessed 09-January-2014].
- [69] ISO/IEC 27002. Iso/iec 27002. [Online; accessed 13-January-2014].

- [70] ISO/IEC 27002:2005. Information technology – security techniques – code of practice for information security management. [Online; accessed 13-January-2014].
- [71] S. Nazir, S. Shahzad, M. Nazir, and H.U. Rehman. Evaluating security of software components using analytic network process. In *Frontiers of Information Technology (FIT), 2013 11th International Conference on*, pages 183–188, Dec 2013.
- [72] Inc. Exceeding Every Exception Technology. Iso/iec 27002:2005 controls and objectives report. [Online; accessed 13-January-2014].
- [73] Common Criteria. Common criteria for information technology security evaluation-introduction and general model. 2012. [Online; accessed 23-July-2013].
- [74] Common Criteria. Common criteria for information technology security evaluation-security functional components. 2012. [Online; accessed 23-July-2013].
- [75] Common Criteria. Common criteria for information technology security evaluation-security assurance components. 2012. [Online; accessed 23-July-2013].
- [76] Open Stack. Certifications and attestations - common criteria. 2014. [Online; accessed 20-March-2014].
- [77] International Society of Automation. Industrial automation and control systems security. [Online; accessed 20-August-2013].
- [78] Cloud Security Alliance Committee. Security guidance for critical areas of focus in cloud computing. 3, 2011.