jamk.fi

Evaluation of Threat Modeling Methodologies

A Case Study

Selin Juuso

Master's thesis
May 2019
School of Technology
Information and Communication Technology
Master's Degree Programme in Cyber Security

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

jamk.fi

Description

Author(s) Selin, Juuso	Type of publication Master’s thesis	Date May 2019
		Language of publication: English
	Number of pages 90	Permission for web publication: x

Title of publication

Evaluation of Threat Modeling Methodologies

A Case Study

Degree programme
Master's Degree Programme in Information Technology

Supervisor(s)
Saharinen, Karo; Saarisilta, Juha

Assigned by
The Finnish Transport and Communications Agency Traficom

Abstract

An interconnected world with an increasing number of systems, products and services relying on the availability, confidentiality, and integrity of sensitive information is vulnerable to attacks and incidents. Unfortunately, the threat landscape expands and new threats, threat agents and attack vectors emerge at all times. Defending against these threats requires that organizations are aware of such threats and threat agents. Threat modeling can be used as part of security risk analysis to systematically iterate over possible threat scenarios.

The focus of the research was on existing threat modeling frameworks and methodologies. Different frameworks were studied in order to discover if any complete and mature enough methodology exists or how the different methods can be combined if necessary. A literature review was used as the primary research method and several different threat modeling methods were studied and summarized. The data gathered from the literature review was verified using a case study consisting of twelve interviews focusing on different threats and threat agents against a system.

The analysis of the responses revealed that there is no comprehensive or complete enough threat modeling methodology available since threat modeling needs are specific to each project and its requirements. While the studied frameworks focus on different topics, e.g. software development, privacy or threat agents, these methods can be combined to cover all the needed aspects. This, however, requires vast knowledge about the different methods available as well as their strengths and weaknesses. This research aims to provide common steps that an organization can take to start modeling threats and give suggestions where more information is available.

Keywords/tags
Threat modeling, threat model, cyber security

Miscellaneous

Appendix 3 is confidential and have been removed from the public thesis. Grounds for secrecy: Act on the Openness of Government Activities 621/1999, Section 24, 7: security arrangements of systems. Period of secrecy is ten years and it ends 18.5.2029.

jamk.fi

Kuvailulehti

Tekijä(t) Selin, Juuso	Julkaisun laji Opinnäytetyö, ylempi AMK	Päivämäärä Toukokuu 2019
		Julkaisun kieli: Englanti
	Sivumäärä 90	Verkkojulkaisulupa myönnetty: x

Työn nimi

Uhkamallinnusmetelmien arviointi

Tutkinto-ohjelma

Master's Degree Programme in Information Technology

Työn ohjaaja(t)

Karo Saharinen

Juha Saarisilta

Toimeksiantaja(t)

Liikenne- ja viestintävirasto Traficom

Tiivistelmä

Verkottotuneessa maailmassa yhä useammat järjestelmät, tuotteet ja palvelut nojaavat arkaluontoisen tiedon saatavuuteen, luottamuksellisuuteen ja eheyteen. Tämä tekee näistä järjestelmistä, tuotteista ja palveluista haavoittuvia erilaisille hyökkäyksille ja vahingoille. Samaan aikaan uhkat monipuolistuvat, niiden määrä kasvaa ja uusia hyökkäysvektoreita ilmestyy jatkuvasti lisää. Suojautuminen näitä uhkia vastaan edellyttää, että organisaatiot osaavat tunnistaa kyseiset uhkat etukäteen. Uhkamallinnus on työkalu, jota voidaan käyttää riskienhallinnan apuna erilaisten uhkien tunnistamiseen.

Opinnäytetyö keskittyy olemassa oleviin uhkamallinnusmetodeihin ja -viitekehyksiin. Erilaisia uhkamallinnustekniikoita läpi käymällä yritettiin löytää metodeja, jotka ovat riittävän monipuolisia ja kattavia vastaamaan erilaisia tarpeita joko yksin tai useita eri metodeja yhdistelemällä. Kirjallisuuskatsauksella kerättiin ensisijainen tutkimusdata, joka vahvistettiin tapaustutkimuksen avulla. Tapaustutkimuksessa käytettiin olemassa olevaa järjestelmää, johon liittyviä uhkia tunnistettiin haastattelujen avulla.

Tuloksia analysoimalla todettiin, että yhtä kaiken kattavaa ja riittävän monipuolista uhkamallinnusmetodia ei ole olemassa, koska tarpeet vaihtelevat huomattavasti ja ovat sidoksissa tiivisti tiettyyn projektiin. Tutkitut menetelmät on kehitetty tiettyyn tarpeeseen, esimerkiksi ohjelmistokehitykseen tai yksityisyyteen liittyvien uhkien tai erilaisten hyökkääjien ja hyökkäystapojen tunnistamiseen. Kuitenkin näitä menetelmiä yhdistelemällä on mahdollista vastata monimutkaisiinkin tarpeisiin, mutta yhdisteleminen vaatii laaja-alaista ymmärrystä eri menetelmistä sekä niiden vahvuuksista ja heikkouksista. Tutkimuksen tavoitteena onkin tarjota yleisiä ohjeita, joiden avulla uhkamallinnuksen voi ottaa organisaatiossa käyttöön sekä viitoittaa lähteitä, joista kiinnostuneet saavat lisätietoa.

Avainsanat

Uhkamallinnus, uhkakartoitus, uhka-analyysi, kyberturvallisuus

Muut tiedot

Liite 3 on salassa pidettävä ja poistettu julkisesta työstä. Salassapidon peruste Julkisuuslain 621/1999 24§, kohta 7: järjestelmän turvajärjestelyt. Salassa pitoaika kymmenen (10) vuotta, salassapito päättyy 18.5.2029.

Contents

Figures

Tables

Acronyms

APT	Advanced Persistent Threat
BIA	Business Impact Analysis
DFD	Data Flow Diagram
EoP	Elevation of Privilege (card game)
MSSP	Managed Security Service Providers
NIST	National Institute of Standards and Technology
PETs	Privacy-enhancing Technologies
SDLC	Software Development Lifecycle
SIEM	Security Incident & Event Monitoring
SOC	Security Operations Center
TTP	Tactics, Techniques & Procedures
UX	User Experience

1 Introduction

Society has become dependent on different systems that process, stores and transmits sensitive information. The omnipresent demand for quick and reliable access to information has made society vulnerable to incidents, attacks and disasters. Retaining the availability, confidentiality and integrity of the information and data is crucial in order to provide both public and commercial services reliably. Simultaneously, new threats, vulnerabilities and exploits emerge and appears at an accelerating pace making software, service and product development, operation and management more laborious and challenging task. Different threat types are spreading to business fields and industries, where manufacturers and organizations are not necessarily prepared to overcome the impacts, ensued from successful attacks or occurred accidents. Changes in the general design of software, service or product are cheaper to make in the early stages of development while fixing bugs, let alone changing the overall architecture just before releasing the product can be expensive and even impossible to accomplish.

The motivation for this research came from the constantly growing need to acquire better tools to tackle the broad and expanding threat landscape present. One identified tool to help to categorize and systematically evaluate the security of a system, product or service, is threat modeling. The purpose of this research is to understand what threat modeling is, how can it be applied and how to find suitable threat modeling methodology for different needs.

Since the thesis work in master’s degree programme should, according to JAMK Master’s degrees’ study guide, be practically and pragmatically oriented applied research or development project that serves the needs of local businesses and organizations and supports regional development, the research consists of two parts: literature review and use case (Master's Thesis - Studyguide). First, research basis, including research theory, research method and research question as well as analysis method, is described. This part also further defines the motivation and objectives of the research.

During the second part of the research, key terminology and different threat types are defined. In addition, the objectives and benefits of the threat modeling process are discussed and different approaches to threat modeling are briefly explained.

Next, the literature review enlightens the current status of existing threat modeling methodologies and frameworks available. Eighteen different methods are studied, evaluated and summarized with the evaluation emphasis on characteristics, such as tailorability, maturity, ease of use and overall focus.

The data gathered from the literature review is then verified by utilizing a case study. This case study contains twelve interviews related to the threat modeling of one particular system. The set of questions asked during the interviews is used to map whether any of the methods studied is sufficient alone for this use case.

Finally, conclusions from the research data are drawn. This section also includes advice for implementing basic threat modeling in an organization. To conclude the research, some ideas that need further development are presented.

2 Research basis

2.1 Research theory

Some of the keywords and concepts linked to the topic of this thesis are first defined. First and foremost, research is a systematic, scientific and scholarly investigation to establish facts or principles and can also be seen as a collection and interpretation of data in an attempt to resolve a problem at hand or to answer a question in a detailed and accurate manner. Applied research is study designed to use its research findings to solve an existing problem and is, therefore, more pragmatic and practical than pure or basic research, which tends to focus more on the fundamental principles and testing of hypothesis for the development of new or revised theories. In any research, data is a necessity, since it consists of the raw facts that record measures of certain phenomena. To conduct proper research work, data should be relevant to the problem the researcher is trying to solve. Data can be either primary data, which is sourced directly by the researcher or secondary data, which tends to be readily available and already in an organized form. Secondary data is usually collected by someone else for some other purpose. Information, on the other hand, is the presentation of facts in a suitable form for the researcher to make decisions. The primary outcomes of the research are called research findings. (Habib, Pathik & Maryam 2014, 3-6).

2.2 Motivation and objectives

Threat modeling methods were first created to assist in the development of more secure operating systems. However, today threat modeling should be an essential part of any risk management process, including also cyber-physical systems. The motivation for evaluating different threat modeling techniques against a specific ICT system comes from working life. A system with a broad threat landscape requires comprehensive threat mapping. Data gathered from threat modeling can be used further in risk management, disaster recovery as well as business continuity planning and crisis management. The primary objective of the research is to find a method to map any serious and probable threats against a critical system.

While some of the existing threat modeling methodologies focus only on software development, some cover just business and/or organizational risks and threats. Some are technical; some are non-technical in nature. Number and type of threats identified will vary significantly, as will the quality, consistency, and the value received from those threat models. Combining these models would probably result in more comprehensive solutions, however, is that enough or even feasible solution?

2.3 Research question

Research questions represent the facets that the researcher wants to know most or first and they help the researcher to have a more focused data set while pointing the researcher towards appropriate data-gathering methods. Research questions may be general or particular, descriptive or explanatory. (Miles, Huberman & Saldaña 2014, 41). A research question can change during the qualitative research; thus Hirsjärvi et al. suggest that the researcher sets research question in more general level. However, if the researcher does not specify a research question at all, the research may end up being just a classification of material. (Hirsjärvi, Remes & Sajavaara 2010, 126)

The main research question for this thesis is: How do existing threat model methods, either alone or together, cover all the different threat types and origins and is there a need for a new, more holistic methodology? The hypothesis behind the research question is that no comprehensive one-size-fits-all method exists, rather different methods or combinations are used in different situations.

2.4 Research method

A research question should dictate the methodological approach used to conduct the research. This research begins with a literature review and is followed by a case study. The purpose of the literature review is to objectively report the current knowledge on a specific topic. The goal for this process in this research is to survey different threat modeling techniques available, find commonalities and differences between them as well as the type of research previously conducted on threat modeling. Hence, in the literature review, the data for the study is collected from published literature. This data from multiple sources is then evaluated and synthesized into a new article, in order to provide all the relevant information about the topic to other researchers in a single paper, thus making information more available. While literature review is used to better understand the subject matter and provide a comprehensive overview, it also helps to place the information into a perspective, as well as determine the different point of views and methods used previously and therefore guide the researcher to generate new research instead of repeating earlier researches. Thus, the researcher should pay attention to the objectivity of the review and reduce the bias often associated with the literature review as much as possible. (Green, Johnson & Adams 2006, 102).

While three basic types of literature reviews exist, a qualitative systematic literature review is used in this thesis. The other two, narrative reviews and quantitative systematic review, also called meta-analyses, are described briefly. Narrative reviews can be either editorials, commentaries or overview articles. Editorials often cover only a few papers and are narrowly focused, while commentaries typically express a particular opinion; thus author’s synthesis demonstrates bias. The research methodology is usually not presented in commentaries. Overview articles or unsystematic narrative reviews are condensed narrative syntheses of each previously published article the author has selected and often they offer a critique on each study. (Green et al. 2006, 103).

In a quantitative systematic review, each paper is critically evaluated, and the results are combined statistically. As stated above, this is also known as a meta-analysis. The goal of this approach is to achieve more objective science from research synthesis by pooling the data between studies in a database and performing appropriate statistics to analyze this large sample data. This can also be a challenge since it might be difficult to find similar enough studies to draw valid conclusions. (ibid., 105)

A qualitative systematic literature review is used to find as many publications as needed to objectively report the existing knowledge on the topic in an organized, detailed, comprehensive and rigorous manner. The author develops a criterion for research publications and based on that criteria decides whether to include or exclude each paper in the final synthesis. The whole process is described in such a transparent way that it is reproducible with the same results afterwards. Each item is reviewed consistently and systematically, and information and details are extracted analogously from the papers. The author should attempt to find also articles that do not support the research hypothesis to reduce the bias further. (ibid., 104)

The topic and objective dictate the depth and breadth of the search strategy for the sources. Selected electronic databases were used as sources of information during this research. The primary source was The Institute of Electrical and Electronics Engineers (IEEE) Xplore Digital Library. Google Scholar, as well as few academic databases, such as MIT Libraries, Finna and Theseus, were also used. Many of the articles found in these sources, had references that led to other related items. All the searches were conducted between November 2018 and March 2019.

Only a few search terms were used in the primary search phase, namely threat model(s) and threat modeling (also a Finnish word for threat modeling was used: uhkamallinnus). This small set of keywords was comprehensive enough to retrieve relevant studies, articles, and literature but narrow enough to focus the effort.

Additional information on each threat modeling methodology was searched using the methodology’s name or abbreviation as a search term (e.g. STRIDE, DREAD, Octave).

The search results were then evaluated with the following selection criteria to include only publications relevant to the topic. First, there were few practical factors: the language of the publication should be either English or Finnish, and the publication should be available for study. The publication should also arise in multiple search sources or be referenced in several other publications. The pertinency of the publication was evaluated with quick browsing of the table of contents and abstracts to find keywords such as threat modeling or the name of the specific threat modeling methodology. More recent publications were emphasized at the expense of the older publications. Each selected publication was studied, and synopsis or summary of different threat modeling methods was formed based on the literature. The summary of the search results can be seen in Appendix 1.

However, research is not just the gathering of information or rearrangement of facts. Any research that produces findings not arrived at by statistical procedures or other means of quantification can be considered as qualitative research. While quantitative analysis usually involves collecting large samples of data and converting them into numerical form for statistical calculations, qualitative research is descriptive and conclusive and usually emphases cases and context, i.e. they engage in a detailed examination of cases related to their chosen topic. It is the collection, analysis, and interpretation of data by observing what people do and say and it refers to the meanings, concepts, definitions, characteristics, metaphors, symbols and descriptions of things. (Habib et al. 2014, 8-9)

Commonly used techniques to collect qualitative data are people-centric methods, such as focus groups and qualitative interviews. Focus group is a method where a selected number of people are brought together to discuss the issue or issues that the researcher is investigating. (Yin 2014, 111). Traditional brainstorming session used in many threat modeling methodologies can also be seen as a focus group type of approach. To get an insight into the experiences and knowledge of the person, qualitative interviews are used. This method provides an opportunity for participants to respond to the issue using their own words; hence new and unexpected perspectives and points of view usually emerge. (Hirsjärvi et al. 2010, 164). In the context of threat modeling, these “known unknowns” are precious assets.

Traditional case study tries to illuminate a decision or a set of decisions: why they were taken, how they were implemented, and with what result. Other common cases include individuals, groups, organizations, processes, programs, neighborhoods, institutions and even events. In other words, a case study allows researchers to focus on a specific “case”, or a contemporary phenomenon, and retain a holistic and real-world perspective as well as investigate the phenomenon in depth and within its real-world context. A case study usually stems from empirical curiosity but is at the same time practical. Therefore it is a good approach when the researcher attempts to understand technology-related processes in an organizational context. A major challenge in case studies involves connecting primary research with the broader theoretical themes and empirical concerns of the existing literature. (Yin 2014, 15-16). Data for the case study is usually collected using multiple different methods such as interviews, observations and research of documents (Hirsjärvi et al. 2010, 134-135).

2.5 Analysis method

Besides dictating the research method, research question(s) should also dictate the methodology to analyze the results. Data gathered from the interviews are condensed to make data stronger. During the data condensations phase, information is simplified, abstracted and transformed from the original interview transcripts. The researcher makes decisions on how the condensation is done, i.e. which data chunks are included or excluded. The goal of data condensation is to sort, focus, organize and sharpen the data to the form that conclusion can be made and verified. (Miles et al. 2014, 31)

Next phase is to display the condensed data in a way that conclusions can be drawn. With qualitative data, the extended text is the most common form. However, humans do not process large amount of information that well, hence other display types, such as matrices, graphs, charts and networks are suggested by Miles et al.

Deciding how the data is displayed, is part of the analysis and analytic activity. (Miles et al. 2014, 32).

Finally, using the data transformed and displayed in previous stages, the researcher tries to find patterns, propositions, explanations and causal flows to draw conclusions and to verify them. It is important to maintain skepticism and openness when making conclusions and verified iterative along the process. The conclusions might be vague at the beginning of the process, but becomes more explicit and grounded as the process iterates. (Miles et al. 2014, 32)

2.6 Previous research

Few researches using either literature review, case study or both to evaluate or study different threat modeling frameworks and methods were examined. Special attention was given to the questionnaires used during the research as well as the overall results.

Dag Eng studied three different threat modeling approaches using textbooks and academic literature and then examined how different techniques can be combined and what advantages and disadvantages there are when using an integrated threat modeling approach. A new method was formed by combining asset-, attacker- and software-centric approaches. This method was then validated with a case study using a hypothetical cloud-based system. (Eng 2017, 4). The technique contains three questionnaires, one for each approach. Each participant answers to only one questionnaire, and all the answers are then extracted and combined with each other to identify threats that appear in all the different approaches.

Sivula (2015) explored different risk and security models in his thesis in order to select a suitable approach for agile health care software development. Different models were evaluated based on literature and questionnaires to identify the best method to use in production.

Launonen (2015) uses Microsoft’s Security Development Lifecycle (SDL) meant for software development to find threats against a factory environment, although the real target of threat modeling still is the IT infrastructure handling the factory’s manufacturing execution system. Threat modeling relies heavily on STRIDE and data flow diagrams.

In his thesis, Holmberg studied threat modeling methodologies based on a literature review and using a case study, evaluated how they fit to be used to analyze security threats against Train Control and Management Systems. A holistic five-step framework based heavily on STRIDE and Boolean logic Driven Markov Processes (BDMP) using attack trees as a base was proposed as a foundation for threat modeling practices. (Holmberg 2016)

3 Threats and threat modeling

3.1 Key terminology

Any organization faces circumstances that can impact and cause harm to the organization's own, other organizations' or even national assets, personnel, processes, mission, function, image or reputation. These circumstances that are potential violations of security are known as threats and are caused by threat sources. (NIST 2012, 8). Any system or environment where the system operates may have both known and unknown vulnerabilities or weaknesses and can be exploited by one or more threats causing a breach of the system's security policy. Since technology continually evolves and changes, new threats and even threat types emerge.

The word threat has an extensive range of different meanings associated with it and it can be understood as people or person, event, weakness or vulnerability and in the context of cybersecurity, also as malware, criminal activity, and espionage. In Vocabulary of Comprehensive Security, a threat is described as an event or a development of events that are possible and harmful. Compared to danger, a threat has a more uncertain evolution phase, and danger is a more practical matter that can be dealt with the risk management procedures. (Finnish Terminology Centre TSK 2017, 40). During the risk management process, threats are usually decomposed further to threat events and threat sources to give a more detailed picture of threats and possible mitigations. In this thesis, a threat is an undesired event or something malicious that can happen to or through a system/product/service.

An asset refers to any resource that has value. The ISO/IEC 27005:2011 standard divides assets to primary assets and supporting assets. Business processes and activities as well as information are considered as primary assets, while assets that the primary assets are relying on, such as hardware, software, network, personnel, site and organization's structure are supporting assets. (ISO/IEC 27005:2011, 2013).

A vulnerability refers to any trust assumption that can be violated to exploit a system; basically, it is a weakness in a system, process, individual, control, implementation, architecture or even organizational structure and external relationship. These weaknesses can be exploited in a harmful way, or they can make an adverse event possible. (Finnish Terminology Centre TSK 2018, 15). In other words, vulnerabilities are revealed when any given threat is dissected and they provide attackers the window of opportunity.

An attack, or exploit, is an action that causes the threat to be realized by utilizing one or more vulnerabilities, while an attack vector is a point or channel that is used to execute the attack. An attack surface refers to a logical or physical area that is exposed to threats and attack patterns. (Souppaya & Scarfone 2016, 4)

A risk is uncertainty or insecurity affecting objectives. Risk causes a deviation from expected and can be positive, negative or both, although the word "risk" is often associated with being implicitly negative. A risk usually contains evaluation of the likelihood and impact and it has a score based on these estimations. (Souppaya & Scarfone 2016, 7)

Countermeasures, controls and mitigations are actions taken to reduce the impacts of a threat or the probability of an attack. Relationships between specific terms can be seen in the chart derived from a graph from Muckin & Fitch (p. 6) in Figure 1.

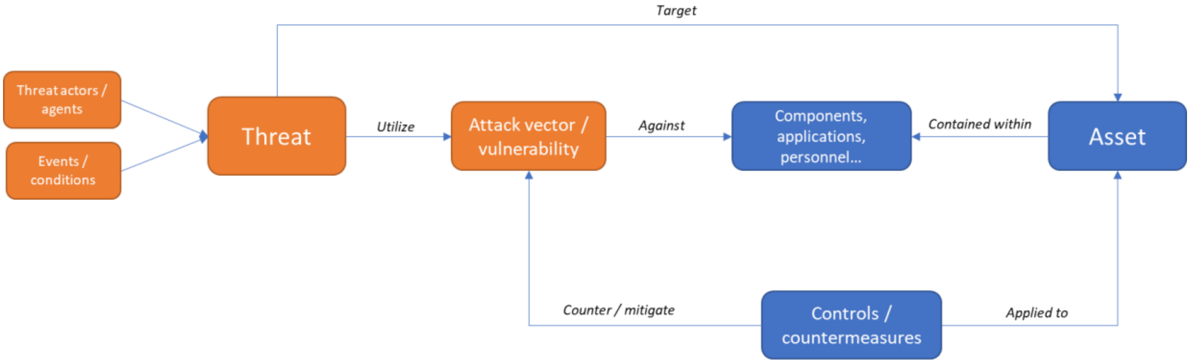


Figure 1. Relationship between different terms

The vocabulary of Comprehensive Security also describes a threat model as a general description of the threats affecting a security environment. Threat models are created to ease and standardize emergency preparedness and emergency planning. Threat assessments are used to define threat models even further. (Finnish Terminology Centre TSK 2017, 41). The National Institute of Standards and Technology (NIST) describes threat modeling as a risk assessment method that is used to model aspects of both offensive and defensive sides of a specific logical entity, which can be a system or an environment, an application or a host or even piece of data or information. (Souppaya & Scarfone 2016, 9).

In practice, threat modeling is used to find security problems using abstractions of the system. Since many of the issues are unique to the specific design of the system, the abstractions layer aids the thinking of the risks towards the bigger picture to find issues and elements that other tools or procedures are unable to find. Abstracting the details when modeling what can go wrong also helps to discover analogies and finding similarities to problems encountered in other systems. Threat modeling has been used traditionally in software security; however, in recent years, threat modeling has been adopted in more complex systems instead of a single application. Today, the threat modeling has been adopted as a part of the security evaluation process in many cases in cyber-physical, autonomous and embedded systems, as well as cloud infrastructure fields.

Threat modeling usually leads to threat scenarios, which are a set of time ordered and discrete threat events attributed to a specific threat source (or multiple sources). Threat scenarios can be represented verbally, graphically or using a tree structure.

In this thesis, threat modeling is a process that includes a different set of techniques to build, design, operate and manage a more secure system, service or product.

3.2 Threats types and threat landscape

The National Institute of Standards and Technology has a taxonomy of describing four threat source types, which are adversarial, accidental, structural and environmental. (NIST 2012, D-2). This categorization is very similar to the one used in the ISO/IEC 27005:2011, where threats are categorized either as deliberate, accidental or environmental. (ISO/IEC 27005:2011 2013, 42). In general, a threat source must have either a method to exploit a vulnerability intentionally or a situation (or method) to exploit vulnerability accidentally. NIST has further defined those four general threat source types as hostile and purposeful attacks; human and machine errors; environmental disruptions, such as disasters and accidents as well as structural failures. Hostile attacks can take place in the cyber or the physical domain, human errors can be a result of omission or commission, and disasters and accidents can be natural or human-made and this category also includes failures of resources that are beyond an organization's control. Structural failures are related to resources, such as software, hardware and environmental controls that the organization has control over. (NIST 2012, 8).

National level actors are often referred to as APTs, or advanced persistent threats, although APTs can also originate from organized crime and other well-resourced groups or these groups can act as a subcontractor for nation level actors. A common characteristic for APTs is an evolving set of TTPs, or tactics, techniques and procedures that are used to establish and maintain a presence in organizations' information infrastructure and exploit the system to exfiltrate information with various purposes. Usually, the objective is to corrupt mission-critical information or to degrade mission capabilities with governmental organizations and espionage with commercial organizations. TTPs are a set of sequential steps performed by adversaries to conduct a cyber attack and are usually characterized by the resources required to be applied and the level of sophistication involved.

As former Secretary of Defense of the United States Donald Rumsfeld responded once in a news briefing (U.S. Department of Defense 2002):

> *Reports that say that something hasn't happened are always interesting to me, because as we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns -- the ones we don't know we don't know. And if one looks throughout the history of our country and other free countries, it is the latter category that tend to be the difficult ones.*

These unknown unknowns – the ones one does not know one does not know – are the reason why threat intelligence should be gathered, and threat modeling should be applied systematically, iteratively and regularly.

3.3 Objectives and benefits of threat modeling

Most of today's business relies on information systems and usually requires new features to be added constantly to the systems. Using threat modeling in the early stages of development will probably decrease the total cost of the project since issues and design flaws that are found early are fixed more easily. As the threats are discovered and triaged at the early stages of development, this also clarifies security requirements and allows devoting resources to a consistent set of features and security properties. As a result, better designs and architectures are created and the need for later re-designing and re-factoring the system due to a constant stream of security bugs found is dramatically lowered. Hence, the final product becomes better and more secure. Additionally, the schedule usually becomes more predictable.

Software, networking, and data elements are continually being added to new physical infrastructures and devices, hence creating new hybrid cyber-physical systems with various new threat types that traditional manufacturers may not understand or even consider as a threat. A growing number of threats and an expansive gap between attack patterns and countermeasures renders reactive mitigation methods more useless. Threat modeling these cyber-physical systems becomes essential in order to find these threats and to produce more secure products. Instead of one-time endeavor, defense strategies become more iterative process because of the dynamic nature of security. Iterative modeling also fosters a greater understanding of likely and emerging new attack sources and methods.

General “best practices” for security have become insufficient since they only cover general set of cases, threats, and vulnerabilities and do not consider the unique characteristics related to each system. Additionally, threat modeling is not a process that can be learned and remembered and then used in several different use cases homogenously. Therefore, threat modeling can elevate organizations’ culture to include a strategic analysis as a fundamental part of any process and develop further the discipline behind the threat analysis. Moreover, threat modeling helps in defining risk mitigation strategies better.

3.4 Threat modeling methods

Different threat modeling methodologies, frameworks, and tools have been developed. Some are more comprehensive than others; some have a higher abstraction level while some focus on a particular domain with greater granularity. Different methods can be distinguished by the logical entity that is being modeled (data, software, system, service, product), the phase of the entity’s lifecycle and the goal of the threat modeling. Threat modeling methods and tools can be consolidated with other methods and even risk management processes to create a custom toolset for special needs.

There are a few general phases in the basic threat modeling method. First, one should understand the logical entity being modeled thoroughly; what does the system/application/host/data/service/product do, how is the data flowing through the entity, where and how is the data stored, who uses the object and so on. The same principles apply to non-technical contexts as well.

3.4.1 Asset- and impact-centric modeling

An asset-oriented approach begins with the identification of critical assets and impacts or consequences towards them. Asset-centric modeling focuses on questions, such as what one’s most valuable assets are and what can go wrong with them. A list of valuable assets is then cycled through, and each asset is considered one at a time. Threat scenarios that can have an impact on the asset are described and prioritized. Assets that have a supporting role or can be used as a stepping stone to harm primary assets should be included. (Shostack 2014, 39).

3.4.2 Attack(er)- and threat-centric modeling

In this approach, potential adversaries and their characteristics, capabilities, resources, intent, relationships and/or behavior are being modeled. Understanding what adversaries desire to gain when attacking against a system, may give an organization more understanding and insight about the tactics, techniques, and procedures (TTP) of the possible adversaries. Adversary behaviors can be organized using a cyber attack lifecycle or cyber kill chain model into a threat scenario or attack scenario.

Threat sources and/or events are usually identified first and threat scenarios and the developments of threats are described in more detail. Adversary characteristics and behaviors as well as intents and motivations are the key elements when identifying impacts. (NIST 2012, 15). Attacker-centric modeling focuses on questions, such as what the attacker wants and why as well as how attackers gain their objectives.

3.4.3 Software- and system-centric threat modeling

Software-centric threat modeling is performed during the software design and development process to reduce vulnerabilities in the software, while system-centric threat modeling focuses on operational systems to improve overall security and tends to be more informal and ad hoc compared to software-centric modeling. (Souppaya & Scarfone 2016, 9). Software-centric modeling focuses on questions, such as what the system is and how it works, as well as what can go wrong and how it can be used incorrectly or harmfully. Hence it is often vulnerability oriented. In software and system-centric modeling techniques, data flow diagrams are usually used to first model the system, data, and boundaries and then determine which threats are relevant to each component and trust boundary crossing.

3.4.4 Data-centric threat modeling

Data-centric threat modeling focuses on protecting particular types of data within a system instead of particular hosts, operating systems or applications. The system and data of interest are identified and characterized and defined narrowly enough. Emphasis is given to the characteristics of authorized locations for storing, transmitting, executing, inputting and outputting data within the system: data flows between authorized locations, security objectives and people and processes authorized to access the data. (Souppaya & Scarfone, 2016).

4 Existing methods and tools

4.1 Attack/threat trees

One of the oldest and most widely used methods for threat modeling cyber-only, physical systems as well as cyber-physical system was developed by Bruce Schneider in 1999. Attack trees or attack graphs are diagrams portraying attacks against a system in a tree form, where the root of the tree is the goal of an attack and the leaves of the three are ways to achieve that goal, as seen in Figure 2. Hence each goal is represented by a separate tree and results in a forest of attack trees. Existing and relevant attack trees can be used to find threats, or an attack tree can be created for a specific use case, however, creating general and multipurpose attack trees is challenging. Each node of the attack tree is iterated and it is analyzed if that issue impacts the system, which is usually modeled with data flow diagrams. (Shostack 2014, 87-88). Today attack trees are often used in combination of with other methodologies. Figure 2 illustrates an attack tree.

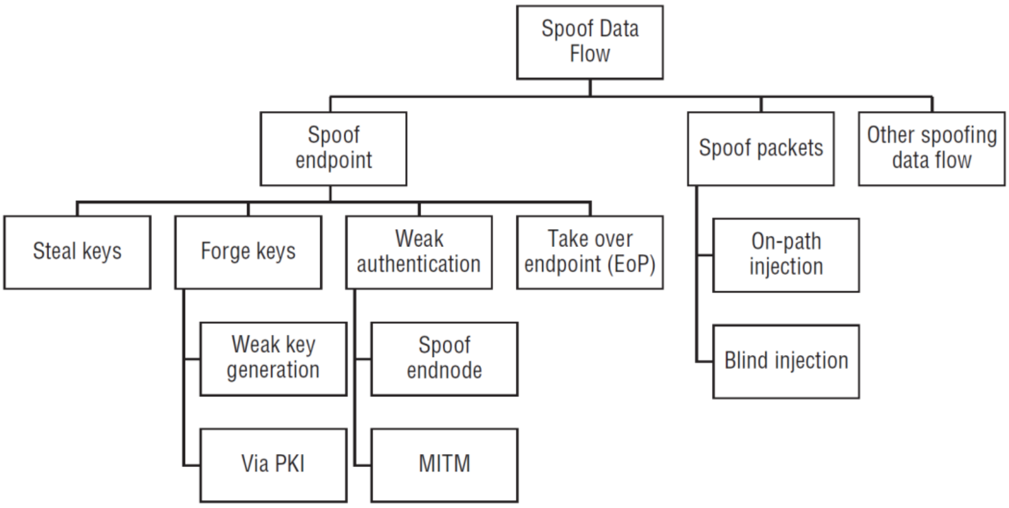


Figure 2. Example of an attack tree (Shostack 2014, 440)

Project-specific trees can be created, and this approach often helps to organize threats better. First, the representation of the three is decided. Trees can be either AND- or OR-trees. A node in an AND-tree is only true if all the nodes below are true, while nodes in an OR-tree are considered true if any of the subnodes below are true. Next, the root node is created. The root node can represent a goal of the adversary or a high-impact action. Subnodes of high-impact actions should state what can go wrong for the node, while subnodes of a root node based on an attacker goal should present different ways the attacker can achieve that goal. Alternative ways to achieve that same goal are presented as unique subnodes. (Eng 2017, 16)

Finally, the completeness of an attack tree is considered; is there a need for additional components, is there anything else that can go wrong or is there any other ways to achieve specific goals? Literature reviews and brainstorming can be used to find additional issues. Additionally, some pruning might be necessary. Each subnode should be analyzed whether it is duplicative or already prevented. Instead of deleting mitigated nodes, they should be marked so they can be ignored during the analysis, however, being still visible, people can see what attacks have already been considered. (Shostack 2014, 88-90)

4.2 STRIDE

STRIDE is a mnemonic and stands for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service and Elevation of Privilege. It is considered to be one of the most mature threat modeling methods. STRIDE was developed by Loren Kohnfelder and Praerit Garg in 1999 and adopted by Microsoft for their internal use to provide more secure software; hence it focuses heavily on software development. STRIDE is not a threat model or threat modeling framework per se but more of a categorization of threats to be considered when developing software. Table 1 describes the different parts of the mnemonic and displays the property being violated.

Table 1. STRIDE explained.

	Description	*Property violated*
Spoofing	Pretending to be something or someone you are not.	Authentication
Tampering	Modifying something you’re not supposed to modify.	Integrity
Repudiation	Claiming you didn’t do something (regardless of whether you did or not).	Non-repudiation
Information disclosure	Exposing information to people who are not authorized to see it.	Confidentiality
Denial of service	Preventing system to provide service by exhausting resources.	Availability
Elevation of privilege	Allowing program or user to do things that they’re not supposed to do.	Authorization

With STRIDE’s approach, the components, system entities, events, trust boundaries and data flows of the system are identified first. Trust boundaries are used to identify such crossing interactions that pose opportunities for attackers. Data flow diagrams (DFD) are drawn to document the system visually. This phase is the most crucial step since the accuracy of the DFDs dictate how successful threat modeling will be. (Shevchenko, Chick, O’Riordan, Scanlon & Woody 2018, 1). A very basic example of a data flow diagram with most basic elements is shown in Figure 3. Any running code is displayed as rounded rectangle or circle and external entities, which are elements outside of an organization's control (such as users or code in browsers and operating systems), are displayed as rectangles with sharp corners. Everything that stores data, i.e. filesystem, database or memory, is displayed with a label and two parallel lines. Communication (data flows) between processes, external entities, and data stores is presented with lines with an arrow showing the direction of the flow. Trust boundaries, drawn as dashed lines, isolate the trustworthy and untrustworthy elements and can represent both logical and physical boundaries. (Shostack 2014, 35).

Mnemonic is then used on found functions and processes, data objects and stores, interfaces and software techniques to find bugs or attack vectors that require mitigations. (Bodeau, McCollum & Fox 2018).

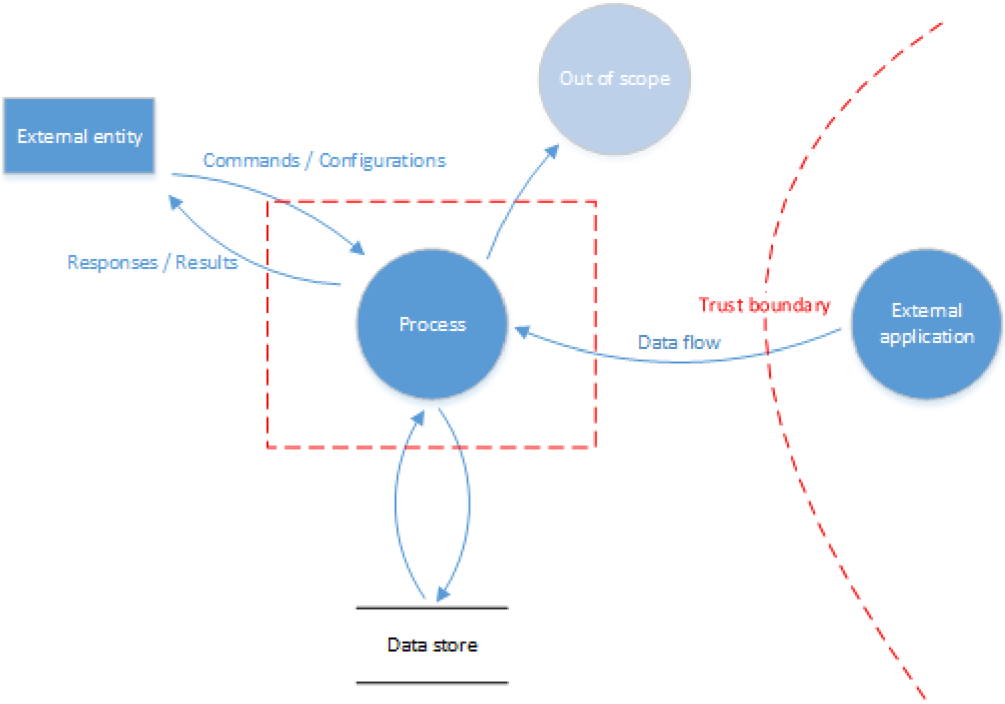


Figure 3. Basic DFD elements

Since STRIDE is primarily intended for analyzing bugs and vulnerabilities in software, getting the best results using STRIDE requires access to the source code. Hence, STRIDE does not fit that well when threat modeling of cyber-physical systems, humans or distributed computing is required, although Khan, McLaughlin, Laverty, and Sezer have presented a comprehensive threat modeling framework for cyber-physical systems using STRIDE in their research. Authors consider STRIDE as a light-weight and effective threat modeling methodology that simplifies the identification of vulnerabilities in cyber-physical systems. (Khan, McLaughlin, Laverty & Sezer 2018).

Scandariato et al. evaluated the performance and productivity of the STRIDE with quantitative observations in laboratory conditions. Their descriptive study proposes that while STRIDE as a threat modeling method is easy to learn and execute and has a low rate of false positives, the rate of false negatives, meaning that many threats are not detected, is moderately high and the modeling process is time-consuming. (Scandariato, Wuyts & Joosen 2013)

There are also few STRIDE variations, two most commonly used of which are STRIDE-per-element and STRIDE-per-interaction. The former approach assumes that certain threat types are more dominant with different DFD elements. This simplifies the threat modeling process since it is easier to focus on the threat types that are most relevant to specific DFD element and threats that are not likely are discarded. The latter approach focuses on data flows between DFD components. Each interaction between components is examined and relevant threats to those interactions are identified. (Eng 2017, 14-16)

4.3 DREAD

DREAD is another acronym created by Microsoft and it stands for discoverability, reproducibility, exploitability, affected users and damage potential. Discoverability determines how easily a vulnerability is detected for a given application environment. Reproducibility helps to determine whether an attack can be successfully repeated. Exploitability defines how easy it is to exploit a known vulnerability and what kind of expertise or resources are needed. Affected users forecast the impact on different types of users using the application, and damage potential evaluates the outcome and overall impact of successful exploitation of a vulnerability. (UcedaVélez & Morana 2015, 167)

This method emphases the use of a scoring system to prioritize and evaluate threats identified by STRIDE or any other methodology by giving each threat a score value of the probability of occurrence in all five categories. Values are then averaged and compared to other averages. The DREAD model may help to determine where most of the effort should be applied to ensure a systems security. Thus, it can be used as a quick risk assessment or analysis tool, since business risks can be illustrated with the viability of an attack and during the risk rating, a modeler can understand the key variables better. (Bodeau et al. 2018)

Although Microsoft has abandoned DREAD from their internal software development toolset in 2010 as too subjective and therefore leading to odd results in many circumstances and omitting many risk factors, DREAD is still used and suggested as an element to threat modeling in many sources. (Shostack 2014, 180). For example, Kaur and Sharma describe a recent threat model where DREAD is applied to get threat ratings and prioritize the relevant threats for an outsourcing business (Kaur & Sharma 2017).

4.4 PASTA

PASTA (or P.A.S.T.A) stands for Process for Attack Simulation and Threat Analysis. The goal of this framework, developed by Tony UcedaVélez in 2012, is to merge technical requirements with business objectives. Although UcedaVélez considers PASTA as a risk-centric framework, it has an attacker-centric perspective, and it produces an asset-centric output. The main goal of the method is to provide a risk mitigation framework that is based on viable threat patterns against various types of mainly unpredictable and sophisticated threats and motives, thus, this step-by-step and iterative approach focuses on understanding business impact, researching threats and developing effective countermeasures as well. PASTA aims to involve all the key decision makers in the process and include security input from multiple domains. (Shevchenko et al. 2018, 4).

PASTA contains seven stages, each of them with multiple activities. The first stage focuses on identification and preparation of risk profiles by defining objectives, such as business, financial and operational objectives, moreover security and compliance requirements are identified and defined. Additionally, Business Impact Analysis (BIA) is conducted, and finally a risk profile is defined. The nature of the first stage requires knowledge about the business processes, financial aspects and governance; hence the primary participants should include project, business and development managers as well as information security officers. (UcedaVélez & Morana 2015, 344-363)

During the second stage of PASTA, the technical scope is defined with five distinct activities in order to understand the underlying technology and map boundaries of the technical environment by identifying dependencies in infrastructure, application, and software. Thus, software components, system-level services, and third-party infrastructure are enumerated, actors and data sinks and sources are identified, and the completeness of secure technical design is asserted. While the primary goal of this stage is to know the system in question thoroughly, the outcome should be a list of the underlying technology stack, including platforms and systems, databases, servers, infrastructure related hardware and any other asset that is used to achieve objectives defined in the first stage. Non-relevant assets should be excluded from the technical scope. Since the second stage is more technical and security-related, participants should include architects, system administrators, engineers, and developers. (UcedaVélez & Morana 2015, 368, 384-387).

The third stage consists of activities aiming to decompose and analyze the application in more detail. First, all use cases of the application are enumerated in order to capture all planned functionalities of the application. This information is then used in the second activity to build DFDs that are understandable by the whole team and include all the essential functions and interactions. In essence, they are an illustration of how different components are interrelated. An example of a data flow diagram of user self-enrollment is presented below in Figure 4.

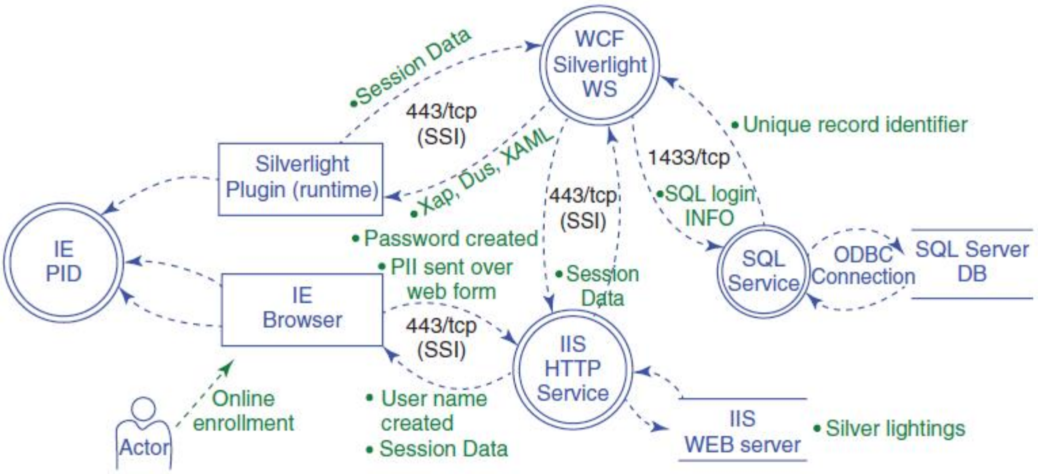


Figure 4. Example of a DFD (UcedaVélez & Morana 2015, 404)

Next, security functional analysis is conducted, and trust boundaries are added to the DFDs. These trust boundaries reveal the areas where new security countermeasures are needed. The application decomposition and dissection stage should include architectural, logical and physical areas in order to be complete: thus, it is imperative to understand how data flows across the application and between components. Therefore architects, developers, and system engineers should be included in this stage. (UcedaVélez & Morana 2015, 393-414).

The actual threat analysis is conducted on the fourth stage, and the key objectives of this stage include reviewing credible and diverse sources of threat data, leveraging internal sources of data, such as logs, alerts and security incidents, enumerating likely threat agents, identifying most likely threats and determining threat likelihood. These objectives are achieved during the six activities described next. First, the overall threat scenario is analyzed by identifying likely threat patterns that are used to target applications with similar architecture, use or technology involved, i.e. list threats against the application data, components, human and physical resources and affiliated infrastructure and applications. Next, threat intelligence is gathered from internal sources, mainly from reported incidents and central log data, including but not limited to alert logs from firewalls and intrusion detection systems, application and server logs, access control logs, database logs, and proxy logs but also human resource and facilities management reports. One good source for threat intelligence is Security Incident and Event Monitoring system (SIEM) in conjunction with the Security Operations Center (SOC). After that, threat intelligence is gathered from external sources, such as threat feeds subscribed from third-party managed security service providers (MSSP) and correlated with the application environment and internal threat intelligence. Threat libraries should then be updated, and PASTA encourages to use libraries such as CAPEC and OWASP. Building and updating an attack library helps to form attack trees and capture the threat to attack relationships. The fifth activity maps threat agents to assets. A threat agent is any individual or group with adversarial intent. The outcome of this stage is a tree-like structure with separate branches for assets, use cases, and threats. Finally, probabilistic values based on considerations for access, opportunity window, the reward for the adversary, simplicity level of the threat and ability to repudiate, are assigned to each identified threat. Incident responders and network operation engineers, or security operation center analysts are key participants and should be included in stage four. (UcedaVélez & Morana 2015, 420-437)

During the fifth stage, weaknesses and vulnerabilities present across the application are identified and analyzed with five specific activities. First, existing vulnerability data is reviewed and correlated. Establishment of a historical context for what vulnerabilities and weaknesses have been found previously in the applications alike gives an excellent starting point. However, vulnerability information should be “fresh”, and too old information should be excluded unless the application under threat modeling is legacy or updates are disregarded due to other issues such as software incompatibilities. The authors suggest that vulnerability data should not exceed twelve months because reviewing old vulnerabilities that might be mitigated with recent patches or updates, makes reviewing vulnerabilities too laborious.

Besides the application itself, the data should also include vulnerabilities relevant to the asset employed by the application, actors, client and server software, third-party software, running services as well as application frameworks, architecture and data sources. Combining the existing vulnerabilities and threat intelligence with historical data and other data collected in previous stages, weak design patterns in the architecture can be identified next. Using data flow diagrams generated during the third stage, data security is reviewed to make sure that appropriate security controls are applied to the data, no matter if it is at rest, in transit or being processed. Then, threats from the attack trees are mapped to vulnerabilities. As a result, abuse cases and vulnerability branches are added to the attack tree; an example of such tree can be seen in Figure 5.

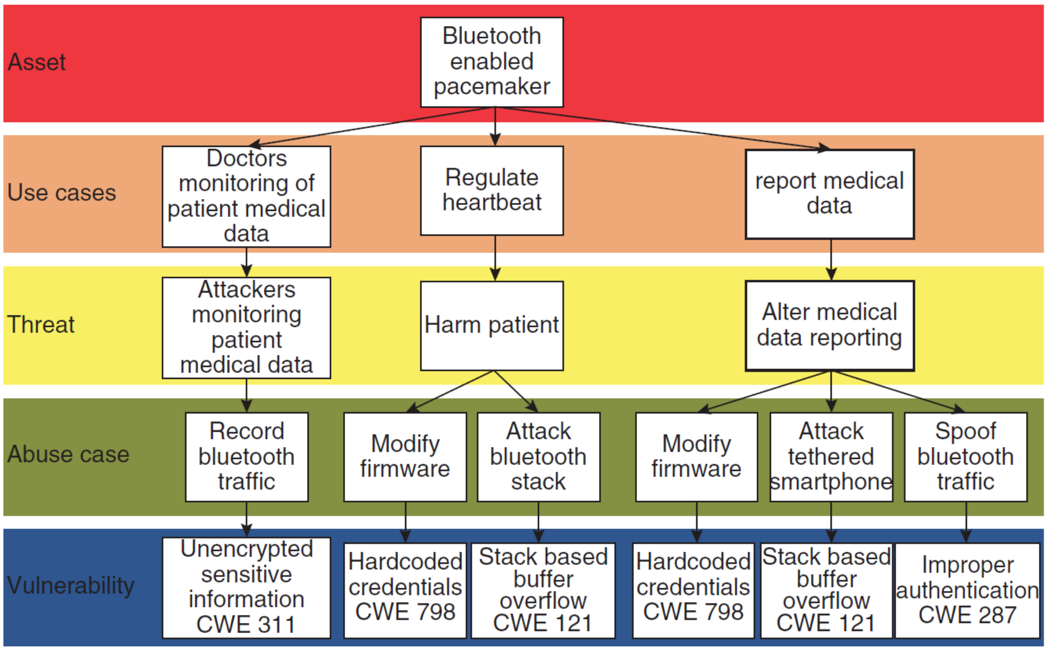

Asset
Bluetooth enabled pacemaker
Use cases
Doctors monitoring of patient medical data
Regulate heartbeat
report medical data
Threat
Attackers monitoring patient medical data
Harm patient
Alter medical data reporting
Abuse case
Record bluetooth traffic
Modify firmware
Attack bluetooth stack
Modify firmware
Attack tethered smartphone
Spoof bluetooth traffic
Vulnerability
Unencrypted sensitive information CWE 311
Hardcoded credentials CWE 798
Stack based buffer overflow CWE 121
Hardcoded credentials CWE 798
Stack based buffer overflow CWE 121
Improper authentication CWE 287

Figure 5. An attack tree (UcedaVélez & Morana 2015, 450)

The abuse cases provide an understandable way to describe high-level attack plans and they are used to map use cases and threats without knowledge about a specific attack vector. Abuse cases are supported by the vulnerability branch to provide a plausible entry point for an attack. The fourth activity in this stage contains a contextual risk analysis that is based on threat vulnerability. Prioritization model for remediation starts to emerge, since identified threats against the most important assets with known vulnerabilities or design flaws receive more attention, while some vulnerabilities or design weaknesses facilitate specific abuse cases. As a result, the possibility of an attack and an association of how attacks can exploit vulnerabilities and design flaws becomes more evident. The fifth and last activity of this stage involves conducting targeted vulnerability testing. To avoid scope creep that is a common problem for vulnerability testing, vulnerability trimming is performed. In this process, only relevant vulnerabilities are selected, based on threat relevance and application components. Several active and passive network and application scanners exist and can be used, and all in-scope asset or component that have not been scanned previously should be included. (UcedaVélez & Morana 2015, 439-456)

The main objective for the sixth stage is to complete the attack tree with attack modeling and simulation. Activities in this stage should be done in parallel to the development phase. First, possible attack scenarios are analyzed further with the enumeration of the threat hierarchy. The process starts from the logical root of the asset and continues through other branches. In the end, a list of possible attack scenarios that are operationally possible and technically feasible given the known vulnerabilities in the environment is formed. Next, the attack library/vectors and control framework are updated to ensure that attack lists and possible control measures are vast enough to build a threat model. Comprehensive and up-to-date attack pattern libraries including a broad range of attack vectors are mandatory as is a list of possible controls. Both should be developed and maintained. Attack patterns are a collection of sequential or non-sequential abusive actions against a target, and each pattern represents a collective of abuse cases for an attack. Authors recommend CAPEC as suitable attack library. If an internal library is being developed, new attack patterns should be added and normalized. Then, the attack surface is identified, and attack vectors are enumerated to finalize attack trees. The attack surface consists of each possible attack that can exploit identified vulnerabilities. The completed threat trees provide a visual representation of the relationships between attacks, vulnerabilities and preceding contributing factors that sustain the attack, such as abuse cases and threats. (UcedaVélez & Morana 2015, 457-468)

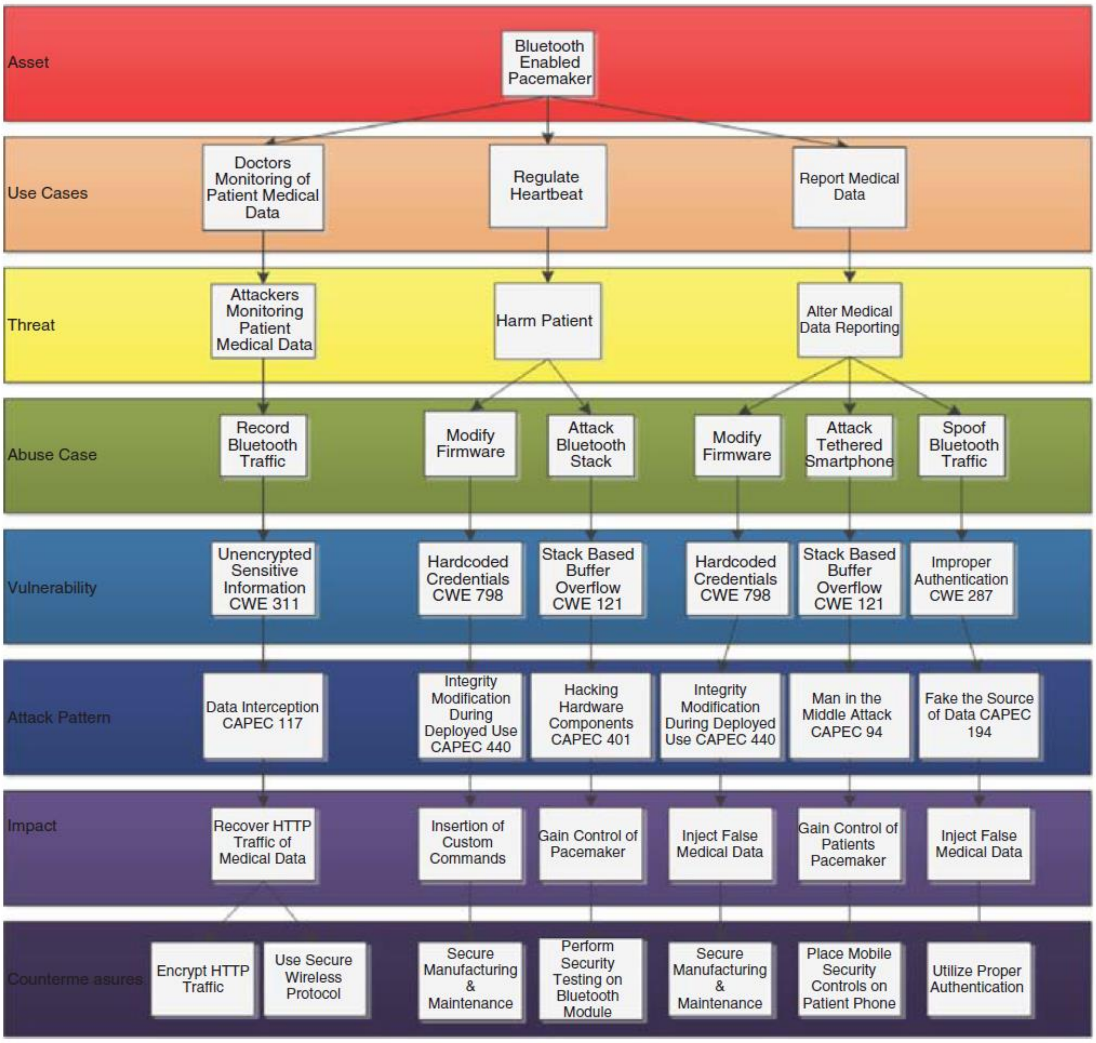
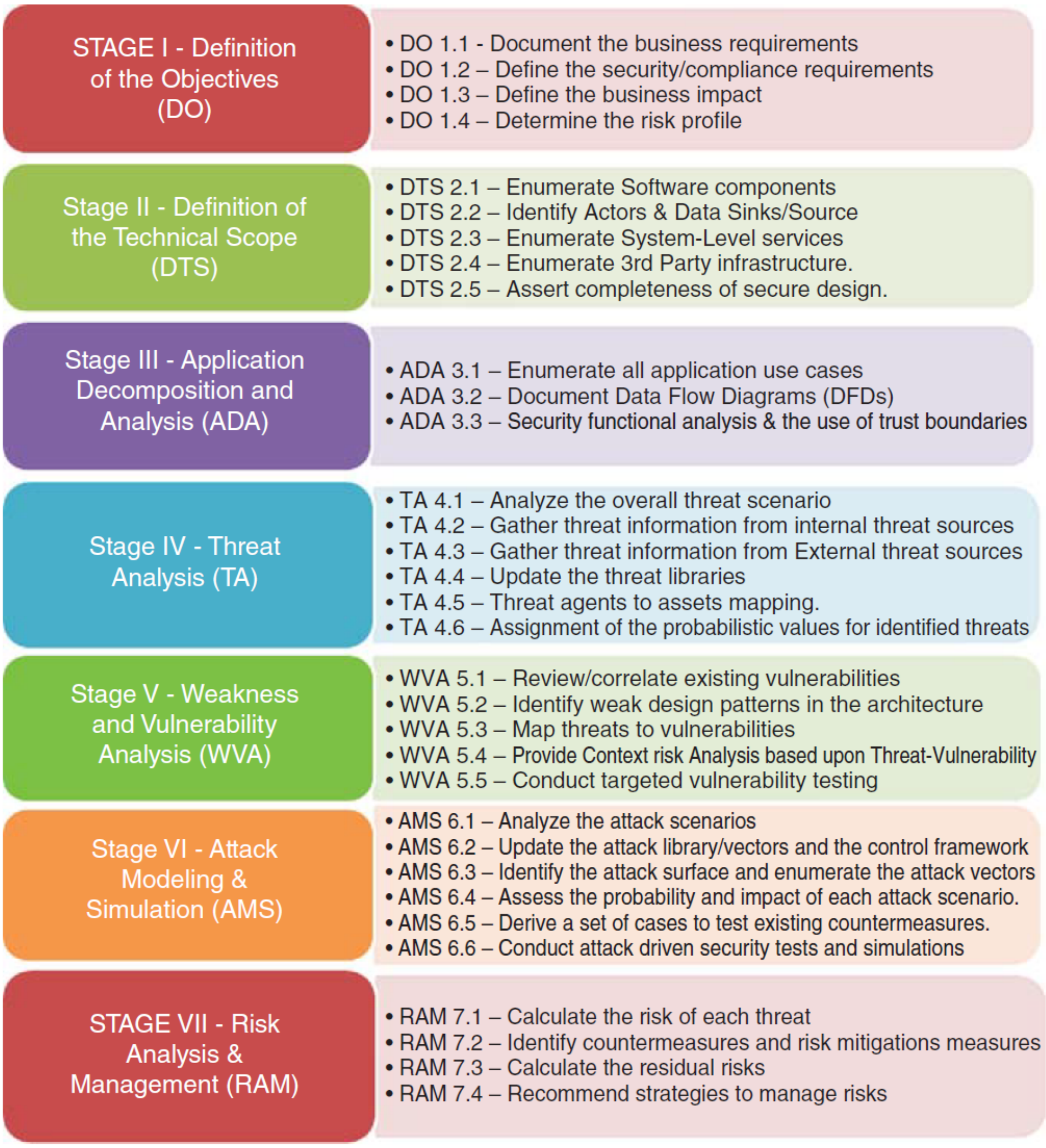

Asset
Bluetooth Enabled Pacemaker
Use Cases
Doctors Monitoring of Patient Medical Data
Regulate Heartbeat
Report Medical Data
Threat
Attackers Monitoring Patient Medical Data
Harm Patient
Alter Medical Data Reporting
Abuse Case
Record Bluetooth Traffic
Modify Firmware
Attack Bluetooth Stack
Modify Firmware
Attack Tethered Smartphone
Spoof Bluetooth Traffic
Vulnerability
Unencrypted Sensitive Information CWE 311
Hardcoded Credentials CWE 798
Stack Based Buffer Overflow CWE 121
Hardcoded Credentials CWE 798
Stack Based Buffer Overflow CWE 121
Improper Authentication CWE 287
Attack Pattern
Data Interception CAPEC 117
Integrity Modification During Deployed Use CAPEC 440
Hacking Hardware Components CAPEC 401
Integrity Modification During Deployed Use CAPEC 440
Man in the Middle Attack CAPEC 94
Fake the Source of Data CAPEC 194
Impact
Recover HTTP Traffic of Medical Data
Insertion of Custom Commands
Gain Control of Pacemaker
Inject False Medical Data
Gain Control of Patients Pacemaker
Inject False Medical Data
Counterme asures
Encrypt HTTP Traffic
Use Secure Wireless Protocol
Secure Manufacturing & Maintenance
Perform Security Testing on Bluetooth Module
Secure Manufacturing & Maintenance
Place Mobile Security Controls on Patient Phone
Utilize Proper Authentication

Figure 6. An example of a completed attack tree (UcedaVélez & Morana 2015, 474)

As seen in Figure 6, each attack tree contains a layer for an asset that can be a server, service, component or data source, use case that reveals the use case associated with the target asset, a threat that describes the planned menace to the asset in high-level, abuse case that provides a counter to the use case's objective, vulnerability that reveals the flaw in business logic, software or design, an attack that depicts the pattern or payload to the target based on vulnerability and impact that describes the outcome of a successful attack. The probability and impact of each attack scenario are assessed to identify the most crucial parts for threat mitigations.

Determining probability can be achieved with the following criteria. Attack prerequisites describe sine qua non conditions, such as how much time a successful attack requires, how complex it is or what it costs to achieve (in means of resources, tools, etc.). Vulnerability maturity characterizes the weakness or vulnerability based on how widely it is disclosed and exploited, how recent the vulnerability is and how much information is available about the vulnerability (proof-of-concept, exploit kits and tools). Hackability defines how easy it is to exploit a weakness or vulnerability either partially or fully. Another way to estimate probabilities is based on contextual information about threat model components and evaluate the probability of threat agents, motives and abuse cases against target assets and maturity of vulnerabilities and rate the severity of the threat, vulnerability and impact. Finally, a set of attack cases is derived in order to test the existing countermeasures; then attack driven security tests and simulations can be conducted to demonstrate attack viability by denoting the impact and probability of the attacks defined in the attack trees. (UcedaVélez & Morana 2015, 457-468)

The seventh and last stage of PASTA focuses on mitigating the relevant threats to application, team or business with activities including calculation of the risk of each threat, identification of countermeasures, calculation of residual risks and formation of recommended strategies to manage risks. Each threat in attack trees should be assigned a percentage weight of the probability based on internal threat data, external threat intelligence and viability of attacks. The right countermeasures are then added to finish the attack trees, and residual risks are calculated with a formula that consisting of threat, vulnerability and impact levels, probability coefficient and the number of countermeasures and the effectiveness of the countermeasures. Finally, the risk profiles associated with the system or application are updated in co-operation with compliance and risk management teams. (ibid. 471-476). The summary of all the stages and activities in PASTA is illustrated in Figure 7.



STAGE I - Definition of the Objectives (DO)	• DO 1.1 - Document the business requirements • DO 1.2 – Define the security/compliance requirements • DO 1.3 – Define the business impact • DO 1.4 – Determine the risk profile
Stage II - Definition of the Technical Scope (DTS)	• DTS 2.1 – Enumerate Software components • DTS 2.2 – Identify Actors & Data Sinks/Source • DTS 2.3 – Enumerate System-Level services • DTS 2.4 – Enumerate 3rd Party infrastructure. • DTS 2.5 – Assert completeness of secure design.
Stage III - Application Decomposition and Analysis (ADA)	• ADA 3.1 – Enumerate all application use cases • ADA 3.2 – Document Data Flow Diagrams (DFDs) • ADA 3.3 – Security functional analysis & the use of trust boundaries
Stage IV - Threat Analysis (TA)	• TA 4.1 – Analyze the overall threat scenario • TA 4.2 – Gather threat information from internal threat sources • TA 4.3 – Gather threat information from External threat sources • TA 4.4 – Update the threat libraries • TA 4.5 – Threat agents to assets mapping. • TA 4.6 – Assignment of the probabilistic values for identified threats
Stage V - Weakness and Vulnerability Analysis (WVA)	• WVA 5.1 – Review/correlate existing vulnerabilities • WVA 5.2 – Identify weak design patterns in the architecture • WVA 5.3 – Map threats to vulnerabilities • WVA 5.4 – Provide Context risk Analysis based upon Threat-Vulnerability • WVA 5.5 – Conduct targeted vulnerability testing
Stage VI - Attack Modeling & Simulation (AMS)	• AMS 6.1 – Analyze the attack scenarios • AMS 6.2 – Update the attack library/vectors and the control framework • AMS 6.3 – Identify the attack surface and enumerate the attack vectors • AMS 6.4 – Assess the probability and impact of each attack scenario. • AMS 6.5 – Derive a set of cases to test existing countermeasures. • AMS 6.6 – Conduct attack driven security tests and simulations
STAGE VII - Risk Analysis & Management (RAM)	• RAM 7.1 – Calculate the risk of each threat • RAM 7.2 – Identify countermeasures and risk mitigations measures • RAM 7.3 – Calculate the residual risks • RAM 7.4 – Recommend strategies to manage risks

Figure 7. Summary of PASTA (UcedaVélez & Morana 2015, 481)

4.5 NIST Special Publication 800-154

The National Institute of Standards and Technology has published a guide for data-centric system threat modeling. The guide is not intended to be a new method, but merely an introduction to data-centric system threat modeling. There are four major steps presented in the publication. First, system and data should be identified and characterized. The identification process should be narrow enough to include only specific data on a specific host or a small group of closely related hosts and devices.

Characterization refers to the process where the system's operation and usage are understood thoroughly. At the minimum level of understanding, authorized locations for the data within the system (such as storage, transmission, execution environment and input/output), how the data travels between these locations, security objectives related to the data and people and processes authorized to access the data are defined. Then, potential attack vectors that could affect negatively any of the identified security objectives for any of the authorized data locations are identified and selected. It is recommended to include all the attack vectors to the model, although this might require too many resources. As a general guideline, criteria for including or excluding attack vectors should be based on the relative likelihood and the impact of a successful attack. (Souppaya & Scarfone, 2016)

During the third phase, security controls for mitigating the attack vectors are characterized. For each selected attack vector, security controls that help to mitigate the associated risk and are feasible to accomplish, are identified and documented. The effectiveness of each control is estimated and ranked with a method that is comparable across mitigations and attack vectors. Additionally, the negative implications, such as increased cost or reduced performance, functionality, and usability, are determined for each control. Finally, the threat model is analyzed. This process verifies the characteristics and documentation from the previous steps and compares all the characteristics together in order to determine how risks can be reduced across all the attack vectors. Overall effectiveness and relative weights are evaluated for each attack vector/security control pair. (Ibid. 11-16)

While this publication describes a qualitative approach to threat modeling primarily, using quantitative approach would provide more precise and accurate results. This, however, requires more resources and would not scale for large and complex systems without massive automation on gathering and analyzing metrics. The qualitative approach provides other benefits. A narrative approach to defining attack vectors can be a flexible way to convey multiple pieces of data in the same content, e.g. the source of malicious content, a vulnerable processor of that content and the nature of the malicious content itself. The narrative approach also makes the documents easier for others outside the threat modeling team to understand attack vectors and risks involved. (ibid. 17-18).

4.6 OCTAVE (Allegro)

OCTAVE, an acronym for Operationally Critical Threat, Asset, and Vulnerability Evaluation, was published in 1999 by Carnegie Mellon Software Engineering Institute and was refined in 2007 to its current version called Allegro. The main goal was to develop a streamlined and optimized process to evaluate information security-related risks with only a small investment of limited resources, such as time and people. The original OCTAVE approach is performed in three phases. First, asset-based threat profiles are created by identifying organizations valuable information-related assets and their current protection strategies, prioritizing assets and selecting most critical assets and documenting their security requirements. Threats interfering those requirements are then identified. During the second phase, the analysis team evaluates information infrastructure and identifies vulnerabilities based on threat profiles created in the previous phase. In the third and last phase, the team develops a security strategy and plans for risk mitigation for critical assets. OCTAVE is primarily intended for large organizations with 300+ employees, multilayered hierarchy and self-maintained ICT infrastructure. However, variations of OCTAVE, such as OCTAVE-S for small organizations, exist. Instead of formal and knowledge eliciting workshops, OCTAVE-S relies on an analysis team with thorough knowledge of threats, essential assets, security requirements and security practices relevant to the organization. OCTAVE-S is also a more structured method and requires a narrower examination of the infrastructure. ((Caralli, Stevens, Young & Wilson 2007, 2-4)

Being an information-centric and risk-based strategic assessment and planning method, OCTAVE Allegro aims to allow more robust risk assessment results with less knowledge about extensive risk assessment. OCTAVE Allegro focuses primarily on information assets; how and where those assets are used, stored, processed and transported, but also how and where these assets are exposed to threats, vulnerabilities, and disruptions. This refined method has eight steps in four stages. The first phase, “Establishing drivers”, includes establishing a set of qualitative measures that build risk measurement criteria consistent with organizational drivers. In the second phase, “Profile assets”, profiles of critical information assets are developed to establish clear boundaries surrounding the asset and to identify asset-related security requirements adequately. The asset profile describes clearly and consistently unique features and qualities of an asset, as well as its characteristics and value. Additionally, the asset containers or locations where assets are stored, transported or processed are identified. A container can be a person, an object or a technology and can reside within the organization's boundaries or out of direct control of the organization. Notably, an asset can have multiple containers. The third phase, called "Identify threats", has two steps. The real-world scenarios, referred to as areas of concerns, are identified, as are different threat scenarios in the context of the locations mapped previously. In other words, possible conditions or situations that can threaten an organization's information asset are identified based on the location of an asset or where the asset "lives"; hence, the number and types of assets involved in the process are limited as well as is the amount of information that needs to be gathered and analyzed. The purpose is not to capture all the possible threat scenarios, but effectively capture situations and conditions that arise naturally from the brainstorming session. The fourth and last phase contains steps for identifying and analyzing risks and selecting mitigation approaches. The risk picture is completed with the consequences to the organization of an identified threat being realized, and all the possible impacts are mapped. Then, a simple quantitative measure is calculated to get a relative risk score for each risk, and finally, a mitigation strategy for each high score risk is developed. (Caralli et al. 2007, 17-20). All the phases and steps described above can be seen in the OCTAVE Allegro roadmap illustrated in Figure 8.

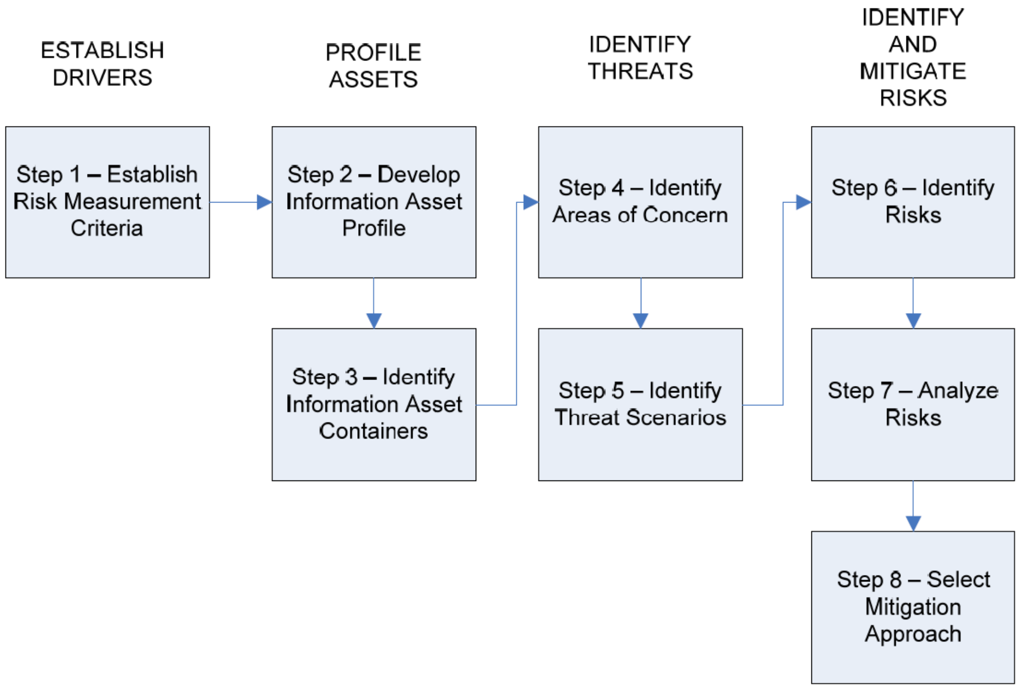

ESTABLISH DRIVERS
PROFILE ASSETS
IDENTIFY THREATS
IDENTIFY AND MITIGATE RISKS
Step 1 – Establish Risk Measurement Criteria
Step 2 – Develop Information Asset Profile
Step 3 – Identify Information Asset Containers
Step 4 – Identify Areas of Concern
Step 5 – Identify Threat Scenarios
Step 6 – Identify Risks
Step 7 – Analyze Risks
Step 8 – Select Mitigation Approach

Figure 8. Stages and steps in Octave Allegro (Caralli et al. 2007, 4)

OCTAVE Allegro provides four threat trees to help threat modelers to consider additional threats: human actors, using technical means, human actors using physical access, technical problems and other problems. Actors, assets targeted or affected by threats, access, means, motives and outcomes, which can be either disclosure, modification, destruction, loss or interruption, are vital attributes of the treat modeling approach in OCTAVE Allegro. (Caralli et al. 2007, 17-20)

4.7 CAPEC

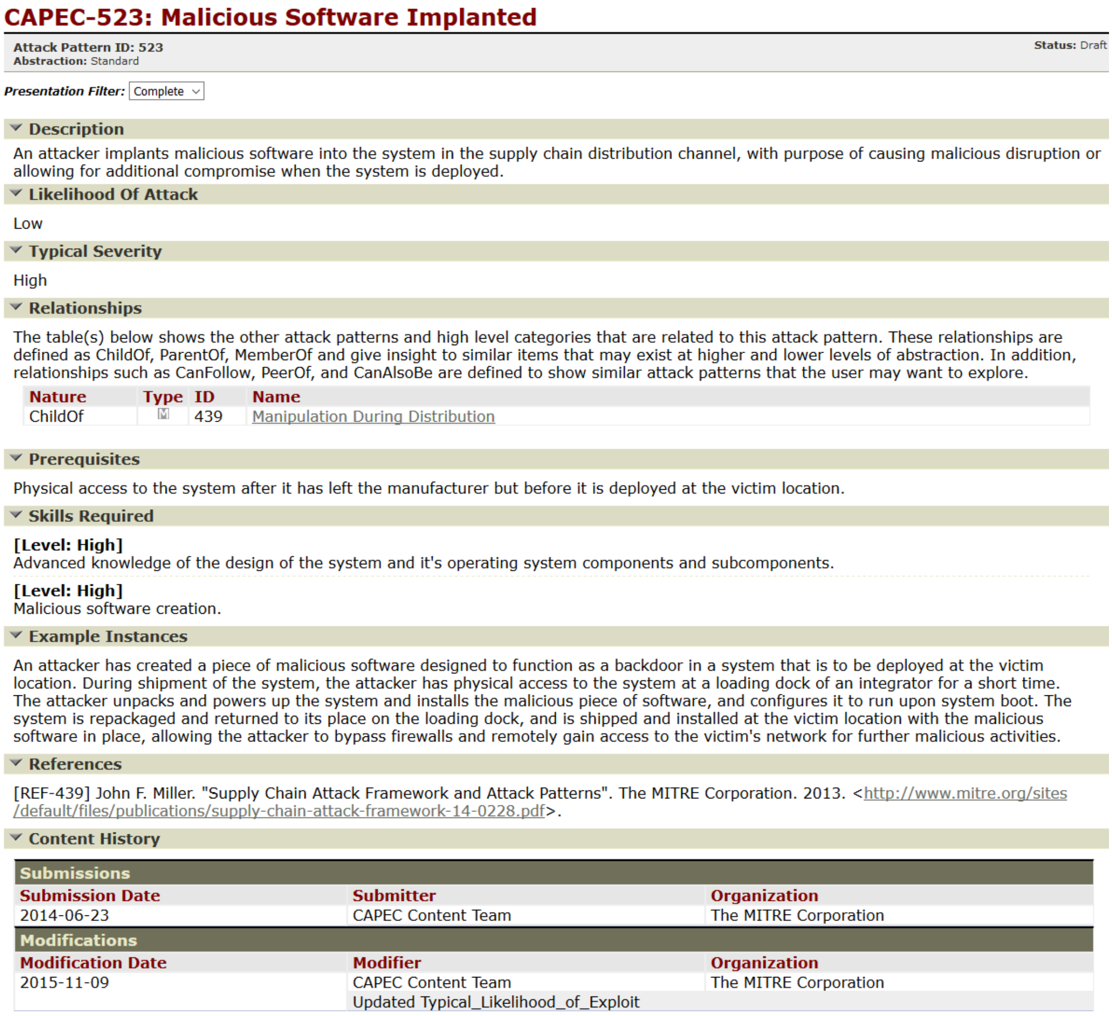
The CAPEC or Common Attack Pattern Enumeration and Classification is a publicly available highly structured set of attack patterns. It was initially established by the U.S. Department of Homeland Security, released in 2007 and resides now under MITRE. Although the CAPEC is not a threat modeling method per se, as a classification of common attacks, it can be used in threat modeling by reviewing the application against CAPEC entries or categories. Since CAPEC has a vast amount of different threats categorized (at the time of writing this, CAPEC had 519 different attack patterns and 49 categories listed on the site), it can also be used to train people about the breadth of the threat landscape. Attack patterns describe common elements and techniques used to exploit weaknesses, therefore helping to understand adversary behavior more widely. (MITRE 2018)

There are three different types of attack patterns: meta, standard and detailed attack pattern. Meta-level patterns describe attack methods or techniques on a more general and higher level and avoid any details on specific implementation or technology. Meta-level patterns are useful when designing the architecture of the system and a generalization of related standard attack patterns. Thus, standard patterns focus on a specific technique or method and aim to provide enough details and information about that technique or method, which often is only a single piece of a fully executed attack. Detailed attack patterns provide even more low-level details on a specific technique with the complete execution flow explained. This execution flow usually contains multiple standard-level attacks chained together. (Ibid.)

CAPEC contains six domains of attack: software, hardware, communications, supply chain, social engineering, and physical security. These domains can be used to find meta-level attack patterns. Attack patterns can also be navigated by attack mechanisms; nine mechanisms existed at the time of writing:

- Engage in Deceptive Interactions
- Abuse Existing Functionality
- Manipulate Data Structures
- Manipulate System Resources
- Inject Unexpected Items
- Employ Probabilistic Techniques
- Manipulate Timing and State
- Collect and Analyze Information
- Subvert Access Control

Besides providing insight on how adversaries may design and execute attacks, attack patterns also recommend methods for mitigating that specific attack. The “file” describing an attack pattern usually contains a description of the attack, likelihood and severity estimations and information about relationships, execution flow, prerequisites, required skills and resources, indicators, consequences, mitigations, example instances, and related weaknesses. (MITRE 2018). Example of an CAPEC attack pattern taken from the CAPEC web site can be seen in Figure 9.



CAPEC-523: Malicious Software Implanted

Attack Pattern ID: 523
Abstraction: Standard
Status: Draft

Presentation Filter: Complete

Description

An attacker implants malicious software into the system in the supply chain distribution channel, with purpose of causing malicious disruption or allowing for additional compromise when the system is deployed.

Likelihood Of Attack

Low

Typical Severity

High

Relationships

The table(s) below shows the other attack patterns and high level categories that are related to this attack pattern. These relationships are defined as ChildOf, ParentOf, MemberOf and give insight to similar items that may exist at higher and lower levels of abstraction. In addition, relationships such as CanFollow, PeerOf, and CanAlsoBe are defined to show similar attack patterns that the user may want to explore.

Nature	Type	ID	Name
ChildOf	M	439	Manipulation During Distribution

Prerequisites

Physical access to the system after it has left the manufacturer but before it is deployed at the victim location.

Skills Required

[Level: High]
Advanced knowledge of the design of the system and it's operating system components and subcomponents.

[Level: High]
Malicious software creation.

Example Instances

An attacker has created a piece of malicious software designed to function as a backdoor in a system that is to be deployed at the victim location. During shipment of the system, the attacker has physical access to the system at a loading dock of an integrator for a short time. The attacker unpacks and powers up the system and installs the malicious piece of software, and configures it to run upon system boot. The system is repackaged and returned to its place on the loading dock, and is shipped and installed at the victim location with the malicious software in place, allowing the attacker to bypass firewalls and remotely gain access to the victim's network for further malicious activities.

References

[REF-439] John F. Miller. "Supply Chain Attack Framework and Attack Patterns". The MITRE Corporation. 2013. <http://www.mitre.org/sites/default/files/publications/supply-chain-attack-framework-14-0228.pdf>.

Content History

Submissions		
Submission Date	**Submitter**	**Organization**
2014-06-23	CAPEC Content Team	The MITRE Corporation
Modifications		
Modification Date	**Modifier**	**Organization**
2015-11-09	CAPEC Content Team Updated Typical_Likelihood_of_Exploit	The MITRE Corporation

Figure 9. An example of CAPEC attack pattern

In summary, CAPEC helps to understand adversary behavior and categorize attacks in a meaningful way to teach designers and developers about various attack mechanisms and mitigation tactics available. Hence, CAPEC is most suitable for application threat modeling. Many of the attack patterns listed by CAPEC are employed by adversaries using techniques described in ATT&CK.

4.8 ATT&CK

Adversarial Tactics, Techniques, and Common Knowledge or ATT&CK is a framework that focuses on network defense and it is based on real-world observations gathered through research, penetration testing and red teaming, threat intelligence reports, conferences and malware samples. The framework was first created in September

2013 by MITRE for a research project and has been publicly available since 2015. The goal of the ATT&CK is to document common tactics, techniques and procedures (TTPs) used by advanced persistent threats (APTs) to target, compromise and operate in an enterprise network in order to provide a large knowledge base of adversarial techniques, that does not focus on the tools or malware but how adversaries interact with the system during an operation. Today, ATT&CK has expanded to incorporate techniques outside the solely Windows-based systems and now has techniques that can be used in multiple operating systems and platforms, such as Linux, macOS, and mobile devices as well as pre-exploit related strategies for planning and conducting operations. (Strom, Applebaum, Miller, Nickels, Pennington & Thomas 2018, 1)

Tactics describe the “why” of a technique and represent the tactical objective of an adversary, i.e. the reason for an adversary to perform a specific action. In the future, MITRE is moving towards one single matrix instead of three different ones; however, today there are three tactics matrices available: Enterprise, Mobile and PRE-ATTA&CK as presented in Table 2.

Table 2. ATT&CK tactics

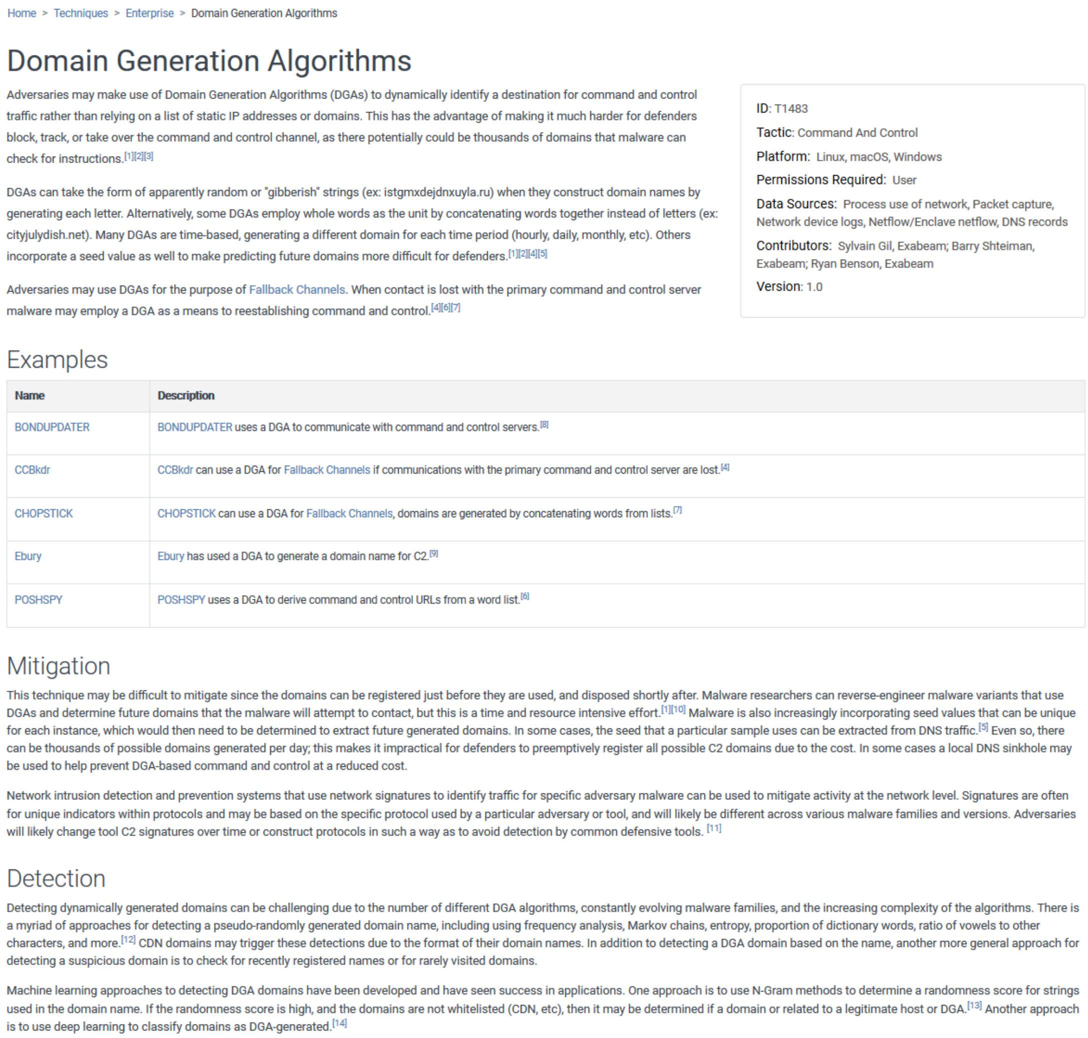
Enterprise Tactics	*Mobile Tactics*	*PRE-ATT&CK Tactics*
Initial Access (10)	Initial Access (9)	Priority Definition Planning (13)
Execution (33)	Persistence (6)	Priority Definition Direction (4)
Persistence (58)	Privilege Escalation (2)	Target Selection
Privilege Escalation (28)	Defense Evasion (8)	Technical Information Gathering (5)
Defense Evasion (63)	Credential Access (11)	People Information Gathering (20)
Credential Access (19)	Discovery (8)	Organizational Information Gathering (11)
Discovery (20)	Lateral Movement (2)	Technical Weakness Identification (9)
Lateral Movement (17)	Effects (6)	People Weakness Identification (3)
Collection (13)	Collection (12)	Organizational Weakness Identification (6)
Exfiltration (9)	Exfiltration (3)	Adversary OPSEC (23)
Command and Control (21)	Command and Control (3)	Establish & Maintain Infrastructure (16)
	Network Effects (9)	Persona Development (6)
	Remote Service Effects (3)	Build Capabilities (11)
		Test Capabilities (7)
		Stage Capabilities (6)

The number inside the parentheses suggests the number of tactics inside a category at the time of writing. The matrix provides a visualized way to see the relationship between tactics and techniques. ATT&CK for Enterprise describes actions that adversaries may take during an intrusion in order to compromise and operate within the enterprise network and focuses on adversary's post-compromise behavior derived from the later stages of Lockheed Martin's Cyber Kill Chain. The enterprise version has three platforms defined (Windows, macOS and Linux).

The ATT&CK for Mobile focuses on threats against mobile devices and other elements of the mobile ecosystem in the mobile environment including two platforms (Android, iOS) and relies heavily on NIST's Mobile Threat Catalogue. Few additional tactics are added compared to the Enterprise-version. For example, network-based effects include tactics and techniques that can be employed without direct access to the mobile device. Differences occur within the common categories between these models, since there are architectural variations, such as multiple radio interfaces and environmental sensors, sandboxes, always-on power-state, and omnipresent network connectivity, between mobile and computer platforms that expose new threats. (Strom et al. 2018, 7)

PRE-ATT&CK expands the tactics beyond technology domains. This section contains hostile behavior during reconnaissance and weaponization prior to the intrusion. If the previous tactics were derived from the later stages of the Cyber Kill Chain, PRE-ATT&CK completes the kill chain by focusing on the first two stages. Tactics, techniques, and procedures that are used to choose a target, gather information and launch a campaign are listed in the PRE-ATT&CK. (Ibid., 8)

Techniques describe "how" adversaries achieve their tactical objectives using an action or "what" adversary gains by performing that action. Many of the techniques described in ATT&CK complement the CAPEC's attack patterns and can be used to educate both red teams or penetration testers and blue teams or defenders. Each technique object contains a name, ID, tactic and description as well as information about the platform, system requirements, required permissions, effective permissions, data source, remote support, defense bypass, CAPEC ID, contributor, examples, detection, and mitigation as seen in the screenshot taken from ATT&CK web site (Figure 10). (Strom et al. 2018, 9)



Home > Techniques > Enterprise > Domain Generation Algorithms

Domain Generation Algorithms

Adversaries may make use of Domain Generation Algorithms (DGAs) to dynamically identify a destination for command and control traffic rather than relying on a list of static IP addresses or domains. This has the advantage of making it much harder for defenders block, track, or take over the command and control channel, as there potentially could be thousands of domains that malware can check for instructions.[1][2][3]

DGAs can take the form of apparently random or "gibberish" strings (ex: istgmxdejdnxuyla.ru) when they construct domain names by generating each letter. Alternatively, some DGAs employ whole words as the unit by concatenating words together instead of letters (ex: cityjulydish.net). Many DGAs are time-based, generating a different domain for each time period (hourly, daily, monthly, etc). Others incorporate a seed value as well to make predicting future domains more difficult for defenders.[1][2][4][5]

Adversaries may use DGAs for the purpose of Fallback Channels. When contact is lost with the primary command and control server malware may employ a DGA as a means to reestablishing command and control.[4][6][7]

ID: T1483
Tactic: Command And Control
Platform: Linux, macOS, Windows
Permissions Required: User
Data Sources: Process use of network, Packet capture, Network device logs, Netflow/Enclave netflow, DNS records
Contributors: Sylvain Gil, Exabeam; Barry Shteiman, Exabeam; Ryan Benson, Exabeam
Version: 1.0

Examples

Name	Description
BONDUPDATER	BONDUPDATER uses a DGA to communicate with command and control servers.[8]
CCBkdr	CCBkdr can use a DGA for Fallback Channels if communications with the primary command and control server are lost.[4]
CHOPSTICK	CHOPSTICK can use a DGA for Fallback Channels, domains are generated by concatenating words from lists.[7]
Ebury	Ebury has used a DGA to generate a domain name for C2.[9]
POSHSPY	POSHSPY uses a DGA to derive command and control URLs from a word list.[6]

Mitigation

This technique may be difficult to mitigate since the domains can be registered just before they are used, and disposed shortly after. Malware researchers can reverse-engineer malware variants that use DGAs and determine future domains that the malware will attempt to contact, but this is a time and resource intensive effort.[1][10] Malware is also increasingly incorporating seed values that can be unique for each instance, which would then need to be determined to extract future generated domains. In some cases, the seed that a particular sample uses can be extracted from DNS traffic.[5] Even so, there can be thousands of possible domains generated per day; this makes it impractical for defenders to preemptively register all possible C2 domains due to the cost. In some cases a local DNS sinkhole may be used to help prevent DGA-based command and control at a reduced cost.

Network intrusion detection and prevention systems that use network signatures to identify traffic for specific adversary malware can be used to mitigate activity at the network level. Signatures are often for unique indicators within protocols and may be based on the specific protocol used by a particular adversary or tool, and will likely be different across various malware families and versions. Adversaries will likely change tool C2 signatures over time or construct protocols in such a way as to avoid detection by common defensive tools. [11]

Detection

Detecting dynamically generated domains can be challenging due to the number of different DGA algorithms, constantly evolving malware families, and the increasing complexity of the algorithms. There is a myriad of approaches for detecting a pseudo-randomly generated domain name, including using frequency analysis, Markov chains, entropy, proportion of dictionary words, ratio of vowels to other characters, and more.[12] CDN domains may trigger these detections due to the format of their domain names. In addition to detecting a DGA domain based on the name, another more general approach for detecting a suspicious domain is to check for recently registered names or for rarely visited domains.

Machine learning approaches to detecting DGA domains have been developed and have seen success in applications. One approach is to use N-Gram methods to determine a randomness score for strings used in the domain name. If the randomness score is high, and the domains are not whitelisted (CDN, etc), then it may be determined if a domain or related to a legitimate host or DGA.[13] Another approach is to use deep learning to classify domains as DGA-generated.[14]

Figure 10. Example of an ATT&CK technique

Besides tactics and techniques, ATT&CK also provides lists of groups and software. The group-list contains details of known adversaries or APT groups that are being tracked by different organizations and security communities. Merely, groups are named intrusion sets, threat or actor groups or campaigns that involve a targeted and persistent threat activity. Many of the groups have multiple names associated since various organizations track the same activity set with different names and group definitions among organizations can be only partially overlapping. Each entry in the group-list contains a name, ID, description, as well as information about aliases, techniques and software group is reported to be using. (Strom et al. 2018, 10-11). At the time of writing, 78 different groups existed in the list.

Software-list consists of tools and applications used by adversaries. There are three higher-level categories: tools, utilities, and malware. Each category can include custom, commercial and open-source code. The list can help analysts to understand how legitimate software can be used to perform hostile actions. Just like with the group, the software can have multiple names. Entries for software-list consist of a name, ID, and description and information about aliases, software type, platform, and related techniques and groups. (ibid., 11-12). At the time of writing, 328 different entries were listed in the software list. Each component or object is related to other components. The high-level representation of the relationship for different ATT&CK objects can be seen in Figure 11.

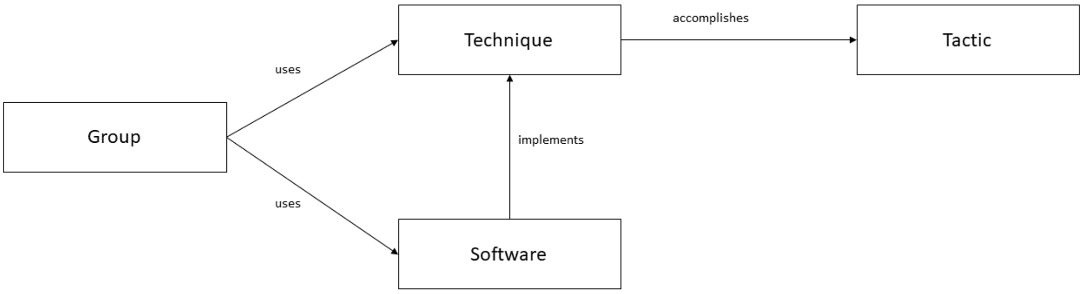


Figure 11. Relationships between ATT&CK objects (Strom et al. 2018, 12)

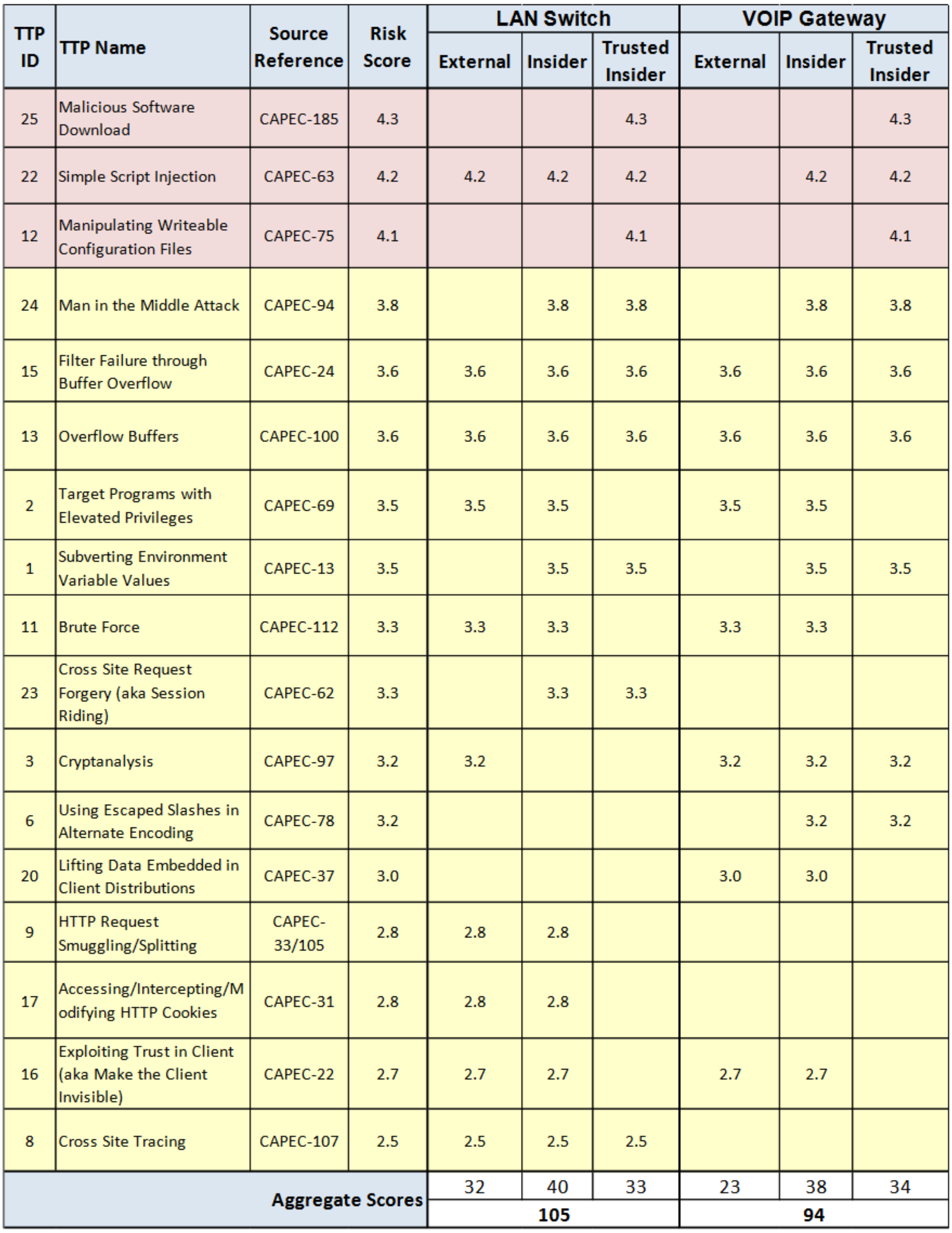
4.9 TARA (MITRE)

Threat Assessment and Remediation Analysis (TARA) is an engineering methodology developed by the MITRE for identifying and assessing cyber threats and determining appropriate and effective countermeasures against those threats to promote greater mission assurance within the system acquisition lifecycle. It is a system level engineering practice within the MITRE Mission Assurance Engineering (MAE) portfolio, which is intended to provide mission assurance against APTs. Hence MITRE’s TARA also focuses on advanced persistent threats.

TARA assessment is used on cyber assets, i.e., any IT asset that either stores, transports or processes information in order to identify and prioritize high-risk adversarial tactics, techniques and procedures (TTPs) these assets might be susceptible to. The approach can be used on both deployed systems and systems that are still in their acquisition phase.

There are three activities and three workflows supported by these activities described in MITRE's technical report: Cyber Threat Susceptibility Analysis (CTSA), Cyber Risk Remediation Analysis (CRRA), and Data and Tools development are the activities, while TARA assessments, catalog development, and toolset development are workflows. TARA assessment uses two different catalogs, one with information about known adversarial TTPs, and the other with its focus on available countermeasures.

The assessment process begins with establishing an assessment scope in order to define and identify which assets and TTPs are evaluated. Usually, the scope also includes the types of adversaries. Then Cyber Threat Susceptibility Analysis is applied to assess how vulnerable each asset is to a range of TTPs. Implausible TTPs are eliminated, and a consistent scoring model between different assessments is applied to the TTPs that cannot be eliminated. This ranking is based on a range of continually evolving criteria, such as impact, likelihood, downtime, restoration costs and level of sophistication. The scoring of the risks helps to set priorities on where to apply security measures to reduce the system's susceptibility to the cyber attack. As a result, from the CTSA, plausible attacks alongside with risk score and adversary type mapped to each cyber asset are listed in a Threat Matrix. Finally, the Threat Matrix is used in Cyber Risk Remediation Analysis to determine countermeasures that will reduce or eliminate the asset's susceptibility to attack. (Wynn, Whitmore, Upton, Spriggs, McKinnon, McInnes, Graubart & Clausen 2011, 1-15). Figure 12 illustrates a threat matrix.



TTP ID	TTP Name	Source Reference	Risk Score	LAN Switch			VOIP Gateway		
				External	Insider	Trusted Insider	External	Insider	Trusted Insider
25	Malicious Software Download	CAPEC-185	4.3			4.3			4.3
22	Simple Script Injection	CAPEC-63	4.2	4.2	4.2	4.2		4.2	4.2
12	Manipulating Writeable Configuration Files	CAPEC-75	4.1			4.1			4.1
24	Man in the Middle Attack	CAPEC-94	3.8		3.8	3.8		3.8	3.8
15	Filter Failure through Buffer Overflow	CAPEC-24	3.6	3.6	3.6	3.6	3.6	3.6	3.6
13	Overflow Buffers	CAPEC-100	3.6	3.6	3.6	3.6	3.6	3.6	3.6
2	Target Programs with Elevated Privileges	CAPEC-69	3.5	3.5	3.5		3.5	3.5	
1	Subverting Environment Variable Values	CAPEC-13	3.5		3.5	3.5		3.5	3.5
11	Brute Force	CAPEC-112	3.3	3.3	3.3		3.3	3.3	
23	Cross Site Request Forgery (aka Session Riding)	CAPEC-62	3.3		3.3	3.3			
3	Cryptanalysis	CAPEC-97	3.2	3.2			3.2	3.2	3.2
6	Using Escaped Slashes in Alternate Encoding	CAPEC-78	3.2					3.2	3.2
20	Lifting Data Embedded in Client Distributions	CAPEC-37	3.0				3.0	3.0	
9	HTTP Request Smuggling/Splitting	CAPEC-33/105	2.8	2.8	2.8				
17	Accessing/Intercepting/Modifying HTTP Cookies	CAPEC-31	2.8	2.8	2.8				
16	Exploiting Trust in Client (aka Make the Client Invisible)	CAPEC-22	2.7	2.7	2.7		2.7	2.7	
8	Cross Site Tracing	CAPEC-107	2.5	2.5	2.5	2.5			
			Aggregate Scores	32	40	33	23	38	34
				105			94		

Figure 12. Example of a threat matrix in TARA (Wynn et al. 2011, 23)

Other workflows, catalog, and toolset development ensure that TTP and countermeasure catalogs and their mappings are up-to-date. The data to these catalogs is derived from both open source and classified sources. (Wynn et al. 2011, 1-15)

4.10 TARA (Intel)

Intel Corporation also has a methodology with an acronym TARA (Threat Agent Risk Assessment) published in December 2009 by Matt Rosenquist. Apart from the acronym, it has nothing in common with the MITRE’s TARA. Intel’s methodology relies heavily on the Threat Agent Library (TAL) initially described in another white paper written by Timothy Casey in 2007. Apart from Rosenquist’s and Casey’s white papers, complete libraries seem to be confidential and are not publicly available.

The main goal of the TARA is to identify most likely attack vectors so that information security strategies can be focused on information security areas of the most significant concern, thus having the highest overall risk rating. According to the white paper, TARA applies specifically to information security and is used merely as a planning tool. In the first phase, all possible threat agents, attacker objectives and attack methods are identified. Threat agents are the origin of risks and are classified based on different characteristics, such as skills, capabilities, resources, intent, and access. Defining their motivations and objectives leads to probable attack methods. When these methods intersect a vulnerability with no mitigations or controls in place the area of exposure is defined. This area of exposure, combined with the possible impact, critical and high-priority areas of concern is formed. (Rosenquist 2009, 3-4)

Intel’s Threat Agent Library (TAL) contains a set or library of threat agents relevant to Intel. There are eight common threat agent attributes and 22 different threat agent archetypes (although there are only 21 presented in the white paper’s matrix; the missing one is probably environmental agents, used by some Intel business units). The characteristics are listed in Table 3. (Casey 2007, 5). In 2015, Intel modified a list of attributes to include motivation and identified ten elements of the motivation parameter (ideology, coercion, notoriety, personal satisfaction, organizational gain, personal financial gain, disgruntlement, accidental, dominance, and unpredictable), and modified its model so that each agent can have multiple motivations (defining motivation, co-motivation, subordinate motivation, binding motivation, and personal motivation). (Bodeau et al. 2018, 22). Table 3 illustrates the threat agent attributes.

Table 3. Threat agent attributes

Attribute	*Variation*	*Description*
Intent	Hostile	harmful and purposeful
	Non-hostile	friendly or accidental
Access (to company's assets)	Internal	
	External	
Outcome	Acquisition/Theft	for resale or extortion
	Business Advantage	to increase the ability to compete
	Damage	towards personnel, assets, information
	Embarrassment	to cause a loss in credibility or brand image
	Technical Advantage	in specific product or production capability
Limits	Code of Conduct	laws and other ethical rules within a profession
	Legal	applicable laws
	Extra-legal, minor	minor and non-violent lawbreaking
	Extra-legal, major	no account of the law
Resource	Individual	independent and average resources
	Club	social or volunteer-based interaction within a group
	Contest	short-lived, maybe anonymous interaction
	Team	a formally organized group with a leader
	Organization	larger and better resourced
	Government	very well resourced and long-term
Skill level	None	no expertise or training
	Minimal	use of existing techniques and tools
	Operational	can create new methods
	Adept	expert in technology and attack methods
Objective	Copy	
	Destroy	
	Injure	
	Take	
	Don't Care	no rational plans or opportunistic objective
Visibility	Overt	an attacker is known before or at the time of execution
	Covert	the attack is noticed but attacker remains unidentified
	Clandestine	attack and the identity of the attacker is meant to be a secret
	Don't Care	no rational plans or has no importance on secrecy

Each agent archetype listed in Table 4 is defined with various attributes. Environmental agents, such as natural disasters and pandemics are used if needed. Each threat agent also has an up-to-date rating based on the agent's recent activity and this rating is updated every six months. (Casey 2007, 5-8). The descriptions for each agent archetype as well as the distinction between hostile and non-hostile agents can be seen in Table 4.

Table 4. Threat agent archetypes

	Agent	*Description*
Non-hostile	Employee, reckless	A current employee who deliberately circumvents safeguards
	Employee, untrained	Current employee unknowingly misuses system or safeguards
	Information Partner	Someone with whom the organization has voluntarily shared sensitive data
Hostile	Anarchist	Rejects all forms of structure and acts with few constraints
	Civil Activist	A highly motivated but non-violent supporter of a cause
	Competitor	Business adversary
	Corrupt Government Official	Using his or her position inappropriately
	Cyber Vandal	Has no strong agenda, but enjoys intrusion and destruction
	Data Miner	An external professional that gathers data
	Employee (disgruntled)	A current or former employee with harmful intents
	Government Spy	State-sponsored insider
	Government Cyberwarrior	A state-sponsored attacker with significant resources
	Internal Spy	Professional that gathers data internally
	Irrational Individual	Someone with illogical and irrational behavior and purpose
	Legal Adversary	An adversary with legal proceedings
	Mobster	Member of organized crime with significant resources
	Radical Activist	Highly motivated, a possible violent supporter of a cause
	Sensationalist	Attention-grabber
	Terrorist	A person with socio-political agenda and violent methods
	Thief	Opportunistic
	Vendor	Business partner

In addition to TAL, TARA also relies on two other libraries: Common Exposure Library (CEL) and Methods and Objectives Library (MOL). CEL enumerates known vulnerabilities and exposures at Intel and maps relevant vulnerabilities against existing controls, while MOL lists known objectives including attacker's motivation and goal, most likely methods leading to these objectives as well as the resulting impact from the successful attack. (Rosenquist 2009, 4)

4.11 IDDIL / ATC

"*There are no idle* (IDDIL) *threats – they attack* (ATC)" is a phrase provided by the creator of this methodology, Lockheed Martin, to help with memorizing the acronym better. The mnemonic contains the following steps:

- Identify Assets
- Define the Attack Surface
- Decompose the System
- Identify Attack Vectors
- List Threat Actors
- Analysis and Assessment
- Triage
- Controls

The IDDIL phase focuses on the discovery, while ATC is considered the implementation phase. During the discovery phase, assets, threats, attacks, and attackers are identified. First, assets of the system that are either business or security assets are identified. Business assets have either data, components or functionality that is crucial for the business operations, while security assets have data, components or functionality that attackers are likely to target. Types and locations of these assets are documented; in addition, current threat intelligence related to these assets is obtained. Next, each asset is scrutinized on macro level to map all the components and elements that contain, communicate or has other access to the asset. This is usually done with data flow diagrams in order to produce an overall image of the attack surface present. With this information, the system is decomposed into a layered view containing more technical details, such as devices, interfaces, protocols, functions, libraries, and APIs. Leveraging the documented and decomposed system and attack surface as well as primary use cases, attack paths and vectors are documented. Each component and functionality areas included in these paths is captured. Additionally, existing security controls and services are included. Finally, potential adversaries, that might want to attack the system, are determined. Attackers' motivations, skill levels, resources, and objectives should be included, and the current threat intelligence should be used as leverage. Figure 13 presents a threat model of a web application including IDDIL elements.

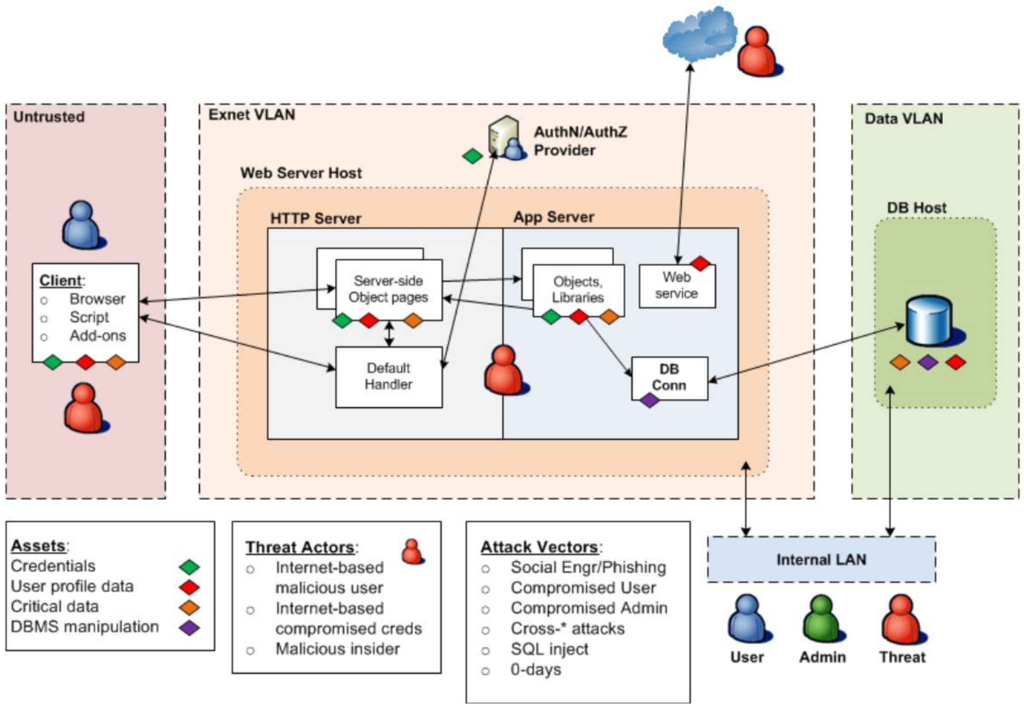


Figure 13. Threat model of a web application (Muckin & Fitch 2015, 17)

Thorough and detailed analysis and assessment are performed in the implementation phase using the data captured in the previous phase. The output of this phase is a prioritized list of items that need to be addressed. Worst-case scenarios, i.e. most likely attacks and the impacts of the successful attacks are determined. The impact on business assets or functionality dictates the triage of the threats, and the impact should have more weight than probability at this point. Finally, the security controls to remove, counter or mitigate the threats are selected and implemented. (Muckin & Fitch 2015, 7-9).

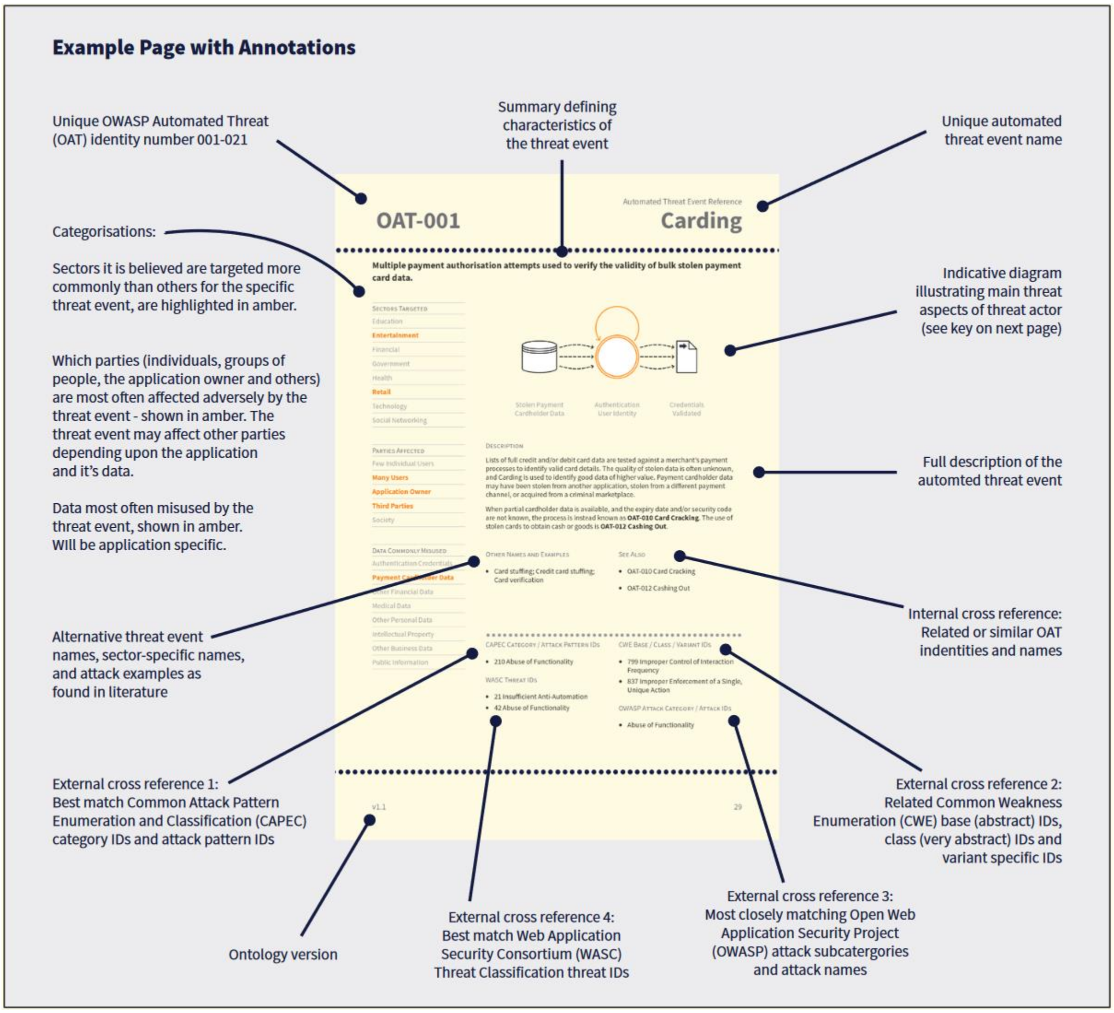
IDDIL/ATC provides a structured process that relies on Lockheed Martin's cyber kill chain and emphases threat intelligence. The process uses attack trees and a variant of STRIDE called STRIDE-LM, which includes the lateral movement to the STRIDE to categorize threats. (Bodeau et al. 2018, 22). A threat profile, a tabular summary of threats, attacks, and related characteristic, is produced to communicate the results of the threat model. A template of the threat profile is presented in Table 5.

Table 5. Threat profile template (Muckin & Fitch 2015, 20)

	DESCRIPTION
ASSET/THREAT OBJECT	The thing the attacker wants or that the owner needs to protect
THREAT TYPES	STRIDE-LM; CIA; Others
ATTACK SURFACE	The components, interfaces, etc that will be initially attacked
ATTACK VECTORS	The path or technique the attacker uses to realize the threat
THREAT ACTORS	The entity who is trying to realize the threat against the asset
RESULTANT CONDITION	Describe what happens if the threat is realized
VULNERABILITIES	Any known vulnerabilities (there may not be any)
CONTROLS	Things that will help mitigate or counter the attack

4.12 OWASP

The Open Web Application Security Project (OWASP) has published a book that describes 21 threat events that are related to web applications and are undertaken using automated actions. The book is called OWASP Automated Threat Handbook, and it provides actionable information and resources to detect and mitigate threats against web applications. The first version was published in July 2015 and the current version (at the time of writing this research, version 1.2) was published in February 2018. As seen in Figure 14, each threat event describes sectors that are more commonly targeted than others, parties that are most often affected by the threat and data types that are commonly misused. Additionally, the description of the threat, cross-reference to other libraries such as CAPEC, WASC and CWE are presented, following with the possible symptoms and suggested threat-specific countermeasures. (Watson & Zaw 2018)



Example Page with Annotations

Unique OWASP Automated Threat (OAT) identity number 001-021

Summary defining characteristics of the threat event

Unique automated threat event name

Categorisations:

Sectors it is believed are targeted more commonly than others for the specific threat event, are highlighted in amber.

Which parties (individuals, groups of people, the application owner and others) are most often affected adversely by the threat event - shown in amber. The threat event may affect other parties depending upon the application and it's data.

Data most often misused by the threat event, shown in amber. WIll be application specific.

Indicative diagram illustrating main threat aspects of threat actor (see key on next page)

Full description of the automted threat event

Alternative threat event names, sector-specific names, and attack examples as found in literature

Internal cross reference: Related or similar OAT indentities and names

External cross reference 1: Best match Common Attack Pattern Enumeration and Classification (CAPEC) category IDs and attack pattern IDs

External cross reference 2: Related Common Weakness Enumeration (CWE) base (abstract) IDs, class (very abstract) IDs and variant specific IDs

Ontology version

External cross reference 4: Best match Web Application Security Consortium (WASC) Threat Classification threat IDs

External cross reference 3: Most closely matching Open Web Application Security Project (OWASP) attack subcatergories and attack names

Automated Threat Event Reference

OAT-001 **Carding**

Multiple payment authorisation attempts used to verify the validity of bulk stolen payment card data.

Sectors Targeted
- Education
- Entertainment
- Financial
- Government
- Health
- Retail
- Technology
- Social Networking

Parties Affected
- Few Individual Users
- Many Users
- Application Owner
- Third Parties
- Society

Data Commonly Misused
- Authentication Credentials
- Payment Cardholder Data
- Other Financial Data
- Medical Data
- Other Personal Data
- Intellectual Property
- Other Business Data
- Public Information

Description

Lists of full credit and/or debit card data are tested against a merchant's payment processes to identify valid card details. The quality of stolen data is often unknown, and Carding is used to identify good data of higher value. Payment cardholder data may have been stolen from another application, stolen from a different payment channel, or acquired from a criminal marketplace.

When partial cardholder data is available, and the expiry date and/or security code are not known, the process is instead known as **OAT-010 Card Cracking**. The use of stolen cards to obtain cash or goods is **OAT-012 Cashing Out**.

Other Names and Examples
- Card stuffing; Credit card stuffing; Card verification

See Also
- OAT-010 Card Cracking
- OAT-012 Cashing Out

CAPEC Category / Attack Pattern IDs
- 210 Abuse of Functionality

WASC Threat IDs
- 21 Insufficient Anti-Automation
- 42 Abuse of Functionality

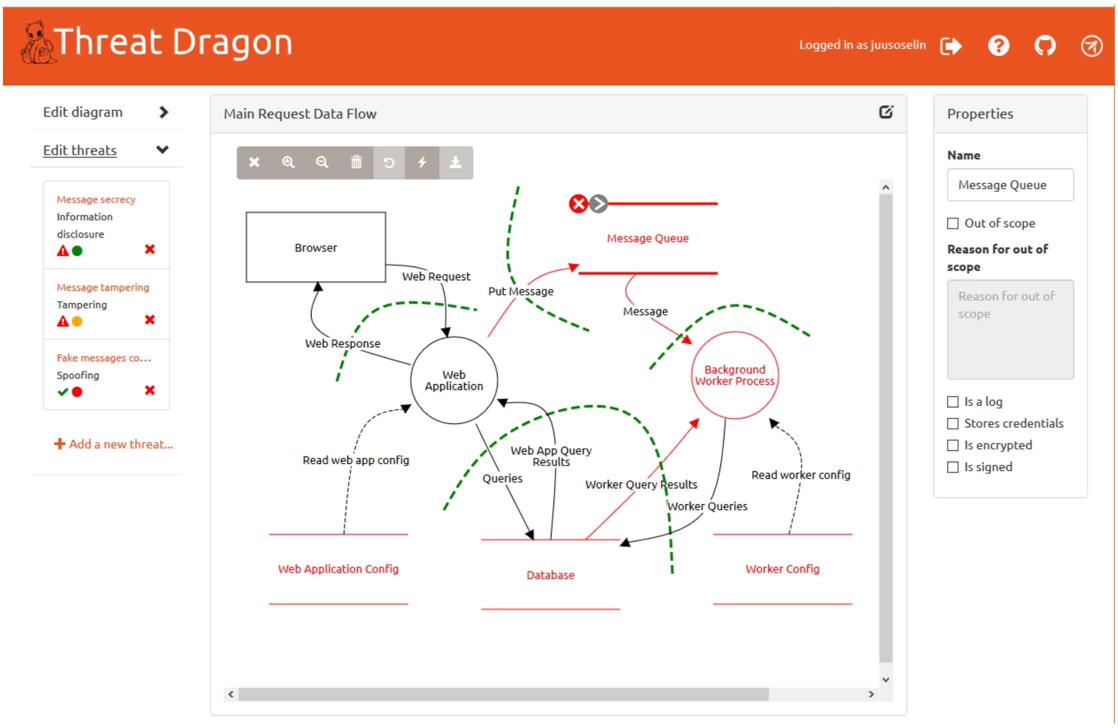
CWE Base / Class / Variant IDs
- 799 Improper Control of Interaction Frequency
- 837 Improper Enforcement of a Single, Unique Action

OWASP Attack Category / Attack IDs
- Abuse of Functionality

v1.1 29

Figure 14. OWASP web application threat legend (Watson & Zaw 2018, 29)

In addition to the handbook, OWASP has other tools worth mentioning when threat modeling is addressed. OWASP Threat Dragon is an open-source online threat modeling web and desktop application to diagram and autogenerates threats and mitigations similar to Mozilla's SeaSponge. (Goodwin 2017). While it is still in early development, it can be in a great assistance when modeling application-based threats. The screenshot of the web application, taken from threatdragon.org, is presented below in Figure 15.



Message secrecy
Information disclosure

Message tampering
Tampering

Fake messages co...
Spoofing

Name
Message Queue

☐ Out of scope

Reason for out of scope

Reason for out of scope

☐ Is a log
☐ Stores credentials
☐ Is encrypted
☐ Is signed

Figure 15. Screenshot of Threat Dragon

Additionally, the OWASP Top Tens are lists of most critical security risks. The list is updated every few years, and therefore it reflects the most prevalent threats at the time being. Consequently, they are a great starting point for a technical threat model. At the time of writing this research, there were Top Ten lists with varying degree of maturity for web applications, Internet of Things (IoT), mobile, privacy, serverless technology, docker container environments, and cloud-native technologies, as well as top five lists for machine learning in early development. OWASP Top Ten for web applications is the most mature list, and each vulnerability describes information about related threat agents and attack vectors, security weaknesses, and impacts. There are also instructions on how to evaluate if an application is vulnerable to this threat, how to prevent it, as well as example attack scenarios. (Bodeau et al. 2018, 28).

4.13 TRIKE

TRIKE is an open source security audit framework that utilizes threat modeling as a technique. It started in 2006 as a standalone desktop application and was later implemented as a spreadsheet. The TRIKE methodology has a defensive perspective and promotes the automation of repetitive tasks in threat modeling in order to provide more resources and focus to risk management. In this approach, modeler first defines the system and builds a requirements model that contains assets, actors, actions, and rules of the system in an Actor-Asset-Action matrix. Each cell is then divided based on CRUD(XF) (creating, reading, updating, deleting, executing, configuring) actions and assigned a value that can be either "allowed action", "disallowed action" or "action with rules". (Saitta, Larcom & Eddington 2005)

Next, data flow diagrams between the elements are built to create an implementation model, which is used to build the actual threat model. Every DFD is iterated through, and related threats are identified. Every unique threat that is discovered becomes a root node in an attack tree. Importantly, TRIKE has only two threat categories: elevation of privilege and denial of service. (Saitta et al. 2005)

Threat model can be then be used to build a risk model, which, although mentioned in the methodology, has not been appropriately implemented to the TRIKE tool. The risk assessment is done with a five-point scale based on the probability of the risk. (Saitta et al. 2005)

As of this writing, TRIKE appears to be no longer maintained (Trike n.d.), however, the analytical and risk-based approach of the TRIKE methodology made this methodology well-known and it is referenced in many sources.

4.14 VAST

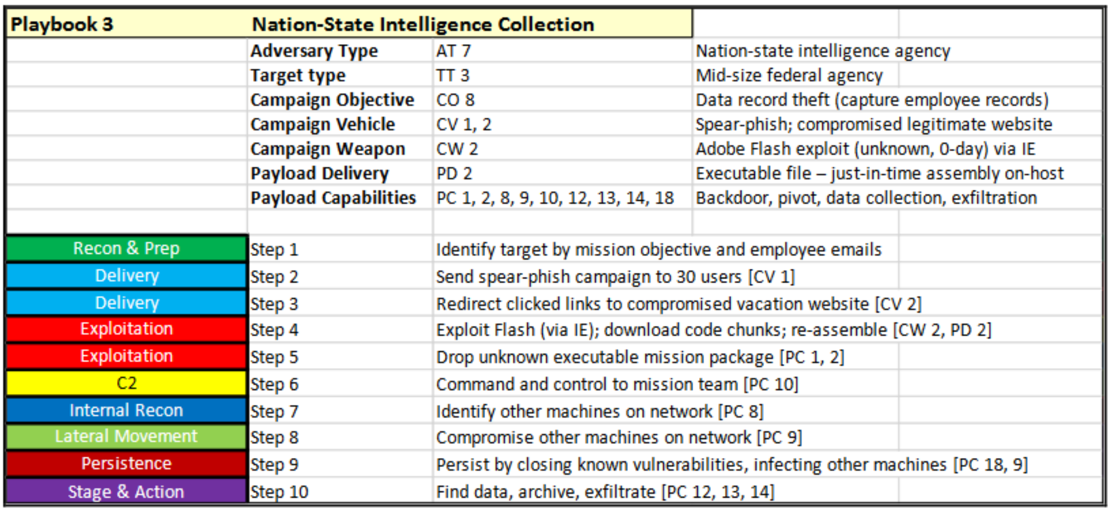
VAST or Visual, Agile and Simple Threat modeling was created by Anurag Agarwal and is based on commercial threat modeling platform, ThreatModeler, which relies heavily on automation. VAST aims to be highly scalable and usable. Thus, large organizations can adopt the method and encircle both their software development lifecycle and entire infrastructure thoroughly and as a result, provide actionable results for different stakeholders. Besides automation, VAST also relies on integration and collaboration. While automation eliminates the repetitive tasks, integration with the tools used in SDLC provides more consistent results and collaboration between key stakeholders produces a more complete, organization-wide view of possible threats and takes advantage of different skill sets available. (ThreatModeler Software, Inc. 2018)

In practice, VAST requires two types of models to be created: threat models for applications and operational threat models. Former use process flow diagrams that represent the architectural point of view, while the latter focuses on attackers' point of view based on the diagrams. (Shevchenko et al. 2018, 16)

4.15 Invincea

Invincea (acquired by Sophos in February 2017) developed an approach to threat modeling carrying the company's name. In the center of this method are adversary and defender playbooks which can be used to run a notional game and identify gaps in the defender's playbook. (Bodeau et al. 2018, 28-29). Invincea's white paper presents a reference adversarial model and sample playbooks as well as defense model and defensive playbooks. The model is extensible to accommodate new adversarial tactics as they evolve with time. According to the paper, knowing the enemy is critical to design proper architecture and manage defenses. Thus, an adversary model, with six attributes, is built first. Each attribute (adversary type, campaign objective, campaign vehicle, campaign weapon, payload delivery, and payload capabilities) has several different options and can be extended as adversaries, and their tactics and objectives evolve. (Invincea 2015)

Next, enterprise security architecture is defined in three primary categories, which are perimeter network defenses, endpoint defenses, and response and recovery. Using the adversarial models, the attacker's playbooks are constructed as seen in Figure 16. Adversaries have inherent advantages over defenders, given their broader freedom to operate. Adversaries can choose the target, the timing of their attack, and the range of tactics used in a particular campaign. (Invincea 2015)



Playbook 3	Nation-State Intelligence Collection		
	Adversary Type	AT 7	Nation-state intelligence agency
	Target type	TT 3	Mid-size federal agency
	Campaign Objective	CO 8	Data record theft (capture employee records)
	Campaign Vehicle	CV 1, 2	Spear-phish; compromised legitimate website
	Campaign Weapon	CW 2	Adobe Flash exploit (unknown, 0-day) via IE
	Payload Delivery	PD 2	Executable file – just-in-time assembly on-host
	Payload Capabilities	PC 1, 2, 8, 9, 10, 12, 13, 14, 18	Backdoor, pivot, data collection, exfiltration
Recon & Prep	Step 1	Identify target by mission objective and employee emails	
Delivery	Step 2	Send spear-phish campaign to 30 users [CV 1]	
Delivery	Step 3	Redirect clicked links to compromised vacation website [CV 2]	
Exploitation	Step 4	Exploit Flash (via IE); download code chunks; re-assemble [CW 2, PD 2]	
Exploitation	Step 5	Drop unknown executable mission package [PC 1, 2]	
C2	Step 6	Command and control to mission team [PC 10]	
Internal Recon	Step 7	Identify other machines on network [PC 8]	
Lateral Movement	Step 8	Compromise other machines on network [PC 9]	
Persistence	Step 9	Persist by closing known vulnerabilities, infecting other machines [PC 18, 9]	
Stage & Action	Step 10	Find data, archive, exfiltrate [PC 12, 13, 14]	

Figure 16. Example of adversarial playbook (Invincea 2015)

Next, the defense teams' playbooks are developed. Since defenders typically do not know who will attack, when, or what tactics are likely to be used against them and they usually must operate with fixed infrastructure and current policies, defender playbooks tend to be more constrained and less agile than adversarial playbooks.

The model is created with a simulated game, where adversaries attack and defenders' responses to see how defenders' playbooks endure against the attacks. This can be achieved by using coverage maps based on defense technology types. Gaps in coverage reveal which attacks are likely successful if executed without any additional security controls. (Invincea 2015)

4.16 LINDDUN

LINDDUN, a mnemonic for Linkability, Identifiability, Non-Repudiation, Detectability, Disclosure of Information, Unawareness, Non-Compliance, is a threat modeling technique focusing on privacy issues and data security and it is used to identify privacy violations in the same manner than STRIDE is used to identify security violations. (Shostack 2014, 121).

Privacy properties, such as anonymity, confidentiality, and unlinkability, are mapped to privacy threats creating the mnemonic, as seen in Table 6. Center of this approach is items of interest (IOI) such as subjects, messages, and actions. Linkability allows an attacker to combine two or more items of interest to distinguish a relation to the specific system, identifiability bundles potential subject to an item of interest, e.g. the sender of a message, thus breaking the anonymity or pseudonymity. Non-repudiation is a threat where an attacker tries to harm a target with credible, but false claims that are hard to counter by the repudiating party. Detectability of an IOI allows an attacker to distinguish if an item exists or not. Disclosure of information is a situation where personal or sensitive information is exposed to parties that are not allowed to have access to it. Content unawareness is a threat where the user provides too much information allowing the attacker to identify the user or the information available is inaccurate, leading to wrong actions by the user. Policy and content noncompliance means that personal and sensitive information can be revealed, although privacy policies are present. Authors also separates privacy properties to hard and soft properties. Hard privacy refers to the possibility to control the privacy violations by providing data as little as possible, hence reducing the need to trust other entities, while soft privacy affiliates to the assumption that user has lost control to personal data and is forced to trust the honesty and competence of data controllers. (Deng, Wuyts, Scandariato, Preneel & Joosen 2011, 7-8). Table 7 also illustrates the distiction between hard and soft properties.

Table 6. Mapping of properties and threats using LINDDUN (Deng et al. 2011, 8)

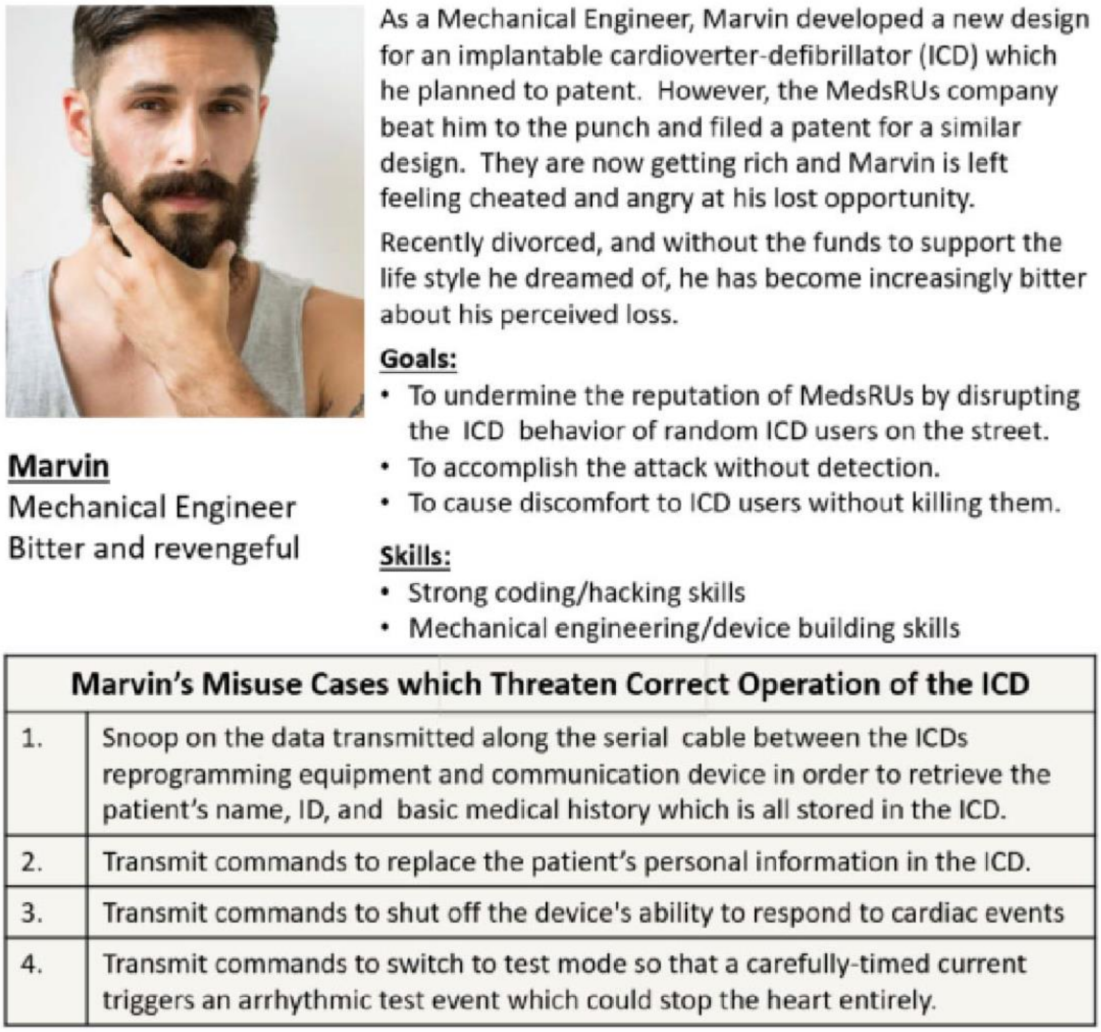
	Properties	*Threats*
Hard	Unlinkability	Linkability
	Anonymity & pseudonymity	Identifiability
	Plausible deniability	Non-repudiation
	Undetectability & unobservability	Detectability
	Confidentiality	Disclosure of information
Soft	Content awareness	Content unawareness
	Policy and consent compliance	Policy and content noncompliance

This method provides a systematic approach to privacy assessment with six steps divided into problem space and solution space. The first three steps (define DFD, map privacy threats to DFD elements and identify threat scenarios) belong to the problem space while following steps (prioritizing threats, eliciting mitigation strategies and selecting corresponding PETs (privacy-enhancing technologies) are steps in the solution space. (Shevchenko et al. 2018, 5)

The first step is analogous to STRIDE’s approach. Threat modeler systematically identifies data flows, data stores, processes and external entities of the system. Then, questionnaires are used to identify threats in the system. Threat categories are mapped to the parts of the system where they may appear and then threat scenarios where these threats may occur, are identified. Following steps are used to find solutions and mitigation strategies against found threats. (Deng et al. 2011)

4.17 Persona non Grata

Persona non Grata (PnG) was developed at DePaul University and is used to model a threat based on motivations and skills of human attackers allowing modelers to visualize threats from the counterpart side. This approach introduces potential attackers or intended archetypal users of the system, their motivations, objectives, and skills to technical experts so that they can identify vulnerabilities and points of compromise relevant to the system. (Shevchenko et al. 2018). Each fictitious persona provides a realistic and engaging representation of a specific user group with traits from psyche, emotions, and background as seen in Figure 17.



As a Mechanical Engineer, Marvin developed a new design for an implantable cardioverter-defibrillator (ICD) which he planned to patent. However, the MedsRUs company beat him to the punch and filed a patent for a similar design. They are now getting rich and Marvin is left feeling cheated and angry at his lost opportunity.

Recently divorced, and without the funds to support the life style he dreamed of, he has become increasingly bitter about his perceived loss.

Marvin
Mechanical Engineer
Bitter and revengeful

Goals:

- To undermine the reputation of MedsRUs by disrupting the ICD behavior of random ICD users on the street.
- To accomplish the attack without detection.
- To cause discomfort to ICD users without killing them.

Skills:

- Strong coding/hacking skills
- Mechanical engineering/device building skills

Marvin's Misuse Cases which Threaten Correct Operation of the ICD	
1.	Snoop on the data transmitted along the serial cable between the ICDs reprogramming equipment and communication device in order to retrieve the patient's name, ID, and basic medical history which is all stored in the ICD.
2.	Transmit commands to replace the patient's personal information in the ICD.
3.	Transmit commands to shut off the device's ability to respond to cardiac events
4.	Transmit commands to switch to test mode so that a carefully-timed current triggers an arrhythmic test event which could stop the heart entirely.

Figure 17. Example of a PnG (Mead, Shull, Spears, Heibl, Weber & Cleland-Huang 2017, 414)

This kind of method with different user types is typically used in user experience (UX) design, where a designer tries to model different ways users behave when using the user interface. (Mead et al. 2017, 412-413). According to Cleland-Huang, creating different hostile personas helps to take a more systematic approach when addressing security concerns throughout a project. (Cleland-Huang 2014). Once adversarial personas are modeled, misuse cases with targets and possible attack mechanisms are identified.

4.18 Security Cards

While STRIDE method usually involves using the Elevation of Privilege (EoP) card game in order to find things that can go wrong more easily and OWASP provides a card game called Cornucopia to help identifying security requirements in Agile, Security Cards refers to the threat brainstorming toolkit developed at the University of Washington. The first version called Control-Alt-Hack was a tabletop card game about white hat hacking and was meant to familiarize computer security related terminology and how security relates to players lives. (University of Washington 2009).

The main purpose of Security Cards is to facilitate the exploration of possible threats and to improve a better security mindset. Hence it is useful also for educators and students. Security Cards is less structured approach and focuses on creativity and brainstorming instead of preconfigured checklists or libraries. The goal is to find more sophisticated and unusual attacks. (Mead, Shull, Vemuru & Villadsen 2018, 5)

The creators of the Security Cards presented the following list of questions that Security Cards should give answers (Denning, Friedman & Kohno 2013):

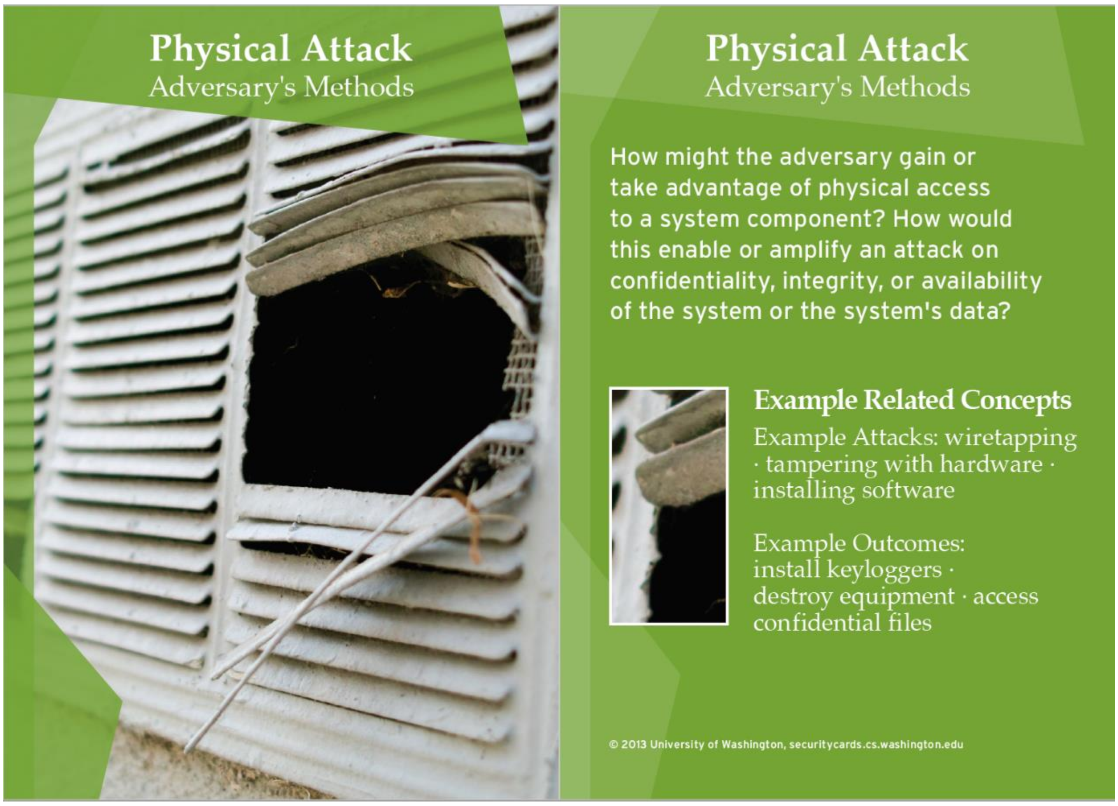
- If your system is compromised, what human assets could be impacted?
- Who might attack your system, and why?
- What resources might the adversary have?
- How might the adversary attack your system?

The deck contains 42 cards and four suits or dimensions: Human Impact (9 cards), Adversary's Motivations (13 cards), Adversary's Resources (11 cards) and Adversary's Methods (9 cards). Human Impacts describes, as the title implies, different ways an attack can impact humans, e.g. violations of privacy or financial loss. Reasons for attacking against a system are covered in Adversary's Motivations, while the tools and expertise needed to accomplish those goals are presented in Adversary's Resources dimension. Adversary's Methods describe how an attack might be carried out. (Mead et al. 2018, 5). All the titles of the cards are presented in Table 7.

Table 7. Card titles

HUMAN IMPACT	**ADVERSARY’S MOTIVATIONS**
• The Biosphere • Emotional Wellbeing • Financial Wellbeing • Personal Data • Physical Wellbeing • Relationships • Societal Wellbeing • Unusual Impacts	• Access or Convenience • Curiosity or Boredom • Desire or Obsession • Diplomacy or Warfare • Malice or Revenge • Money • Politics • Protection • Religion • Self-Promotion • World View • Unusual Motivations
ADVERSARY’S RESOURCES	**ADVERSARY’S METHODS**
• Expertise • A Future World • Impunity • Inside Capabilities • Inside Knowledge • Money • Power and Influence • Time • Tools • Unusual Resources	• Attack Cover-Up • Indirect Attack • Manipulation or Coercion • Multi-Phase Attack • Physical Attack • Processes • Technological Attack • Unusual Methods

Each card contains the topic, dimension and an evocative photograph on one side and illustrative examples and questions for clarification and to jumpstart thinking on the other side as seen in Figure 18. (University of Washington 2013).



Physical Attack
Adversary's Methods

Physical Attack
Adversary's Methods

How might the adversary gain or take advantage of physical access to a system component? How would this enable or amplify an attack on confidentiality, integrity, or availability of the system or the system's data?

Example Related Concepts

Example Attacks: wiretapping · tampering with hardware · installing software

Example Outcomes: install keyloggers · destroy equipment · access confidential files

© 2013 University of Washington, securitycards.cs.washington.edu

Figure 18. Example of a security card

The card game is supposed to be played in groups of 3-5 members. Each group goes through the deck of cards to familiarize with the dimensions and the cards. Then cards are arranged based on how relevant each topic is to the system being modelled and how severe overall risk do they present. The reasons why the cards were arranged in that specific order are then discussed and possible attack scenarios and attacker profiles are documented. (Denning et al. 2013)

4.19 Summary

The literature review confirmed the vast amount of different threat modeling techniques and methods available. All the methods have been originally developed for specific purpose and this limits the usability on different cases. However, with creativity and modification, most of these methods can be used as a starting point. Described methods were evaluated based on their focus, maturity, tailorability and ease of use, including the level of documentation and overall learning curve. The methods that shine in one area, usually succeeds in other areas as well. STRIDE, for example, is perhaps the most mature method available and is easy to learn and execute. However, it’s tailorability depends heavily on modeler’s creativity, since it is mainly suitable for software development. Pros and cons of each evaluated framework are collected into the Table 8 below.

Table 8. Summary of different methods and frameworks

	Pros	*Cons*
Attack trees	• Easy to adopt and understand (when using existing trees) • Consistent results with repetition • Can be used separately for each component instead of building a complex system as a whole	○ Requires thorough understanding of the system and high expertise in cyber security ○ New and generic attack trees are very hard to create ○ Does not provide guidelines for assessing sub-goals, attacks, or risks
STRIDE	• Easy to learn and execute • One of the most mature methods • Good documentation	○ Mainly suitable for software development ○ Number of threats can grow rapidly as a system increases in complexity ○ Time and resource intensive
DREAD	• Contributes to risk management • Simple	○ Many of the risk factors are missing ○ More or less deprecated
PASTA	• Elevates threat modeling to a strategic level • Has built-in prioritization of mitigation • Excellent documentation	○ Laborious and extensive process
NIST SP 800-154	• Easy to understand and adopt	○ Still a draft
OCTAVE	• Main aspects are operational risk, security practices, and technology • In-depth and flexible method • Contributes to risk management • Has built-in prioritization of mitigation • Consistent results with repetition • Scalable	○ Vague and large documentation ○ Time consuming process ○ Focuses on organizational risks and does not address technological risks ○ Evaluates activities, not continuous processes
CAPEC	• Impressive size and scope • Includes an assessment of completion for each entry • Constantly updated	○ Very large library, thus time consuming and might be intimidating to start using
ATT&CK	• Impressive size and scope • Constantly updated • Hunting for new threats enhances threat intelligence	○ Doesn't focus on individual threats ○ Focuses mainly on advanced persistent threats

TARA (Mitre)	• Tools can be omitted or tailored to suit the needs • Provides default scoring tools to assess risk	○ Focuses information security only and mainly on advanced persistent threats
TARA (Intel)	• TAL is easy to adopt • Attacker-centric approach can be useful in many cases	○ Focuses on information security only ○ Libraries are not public
IDDIL / ATC	• Structured process that is easy to learn	○ Limited documentation
OWASP	• Constantly updated • Easy to adopt	○ Development relies heavily on single individuals or groups ○ Focuses on only some subsets of threats in specific environment
Trike	• Contributes to risk management • Has built-in prioritization of mitigation • Automation	○ Vague and insufficient documentation ○ Unfinished, since the development has stopped ○ More or less deprecated
VAST	• Contributes to risk management • Has built-in prioritization of mitigation • Consistent results with repetition • Automation • Scalable	○ Vague and insufficient documentation ○ Commercial
Invincea	• Gamification	○ Focuses mainly on security products as controls ○ Limited documentation
LINDDUN	• Extensive privacy knowledgebase and documentation • Has built-in prioritization of mitigation	○ Steep learning curve (unusual terminology) ○ Labor intensive and time consuming ○ Number of threats can grow rapidly as a system increases in complexity
PnG	• Has built-in prioritization of mitigation • Consistent results with repetition	○ Very limited (only some subsets of threats)
Security Cards	• Gamification • Good for education • Emphasis on creativity • Wide range of threats included	○ Consistent results vary too much between teams ○ The number of false positives may become high

5 Case study

5.1 Interviews

According to Yin, interviews are an essential part of any case study. The case study entails an ICT system with a complex architecture and a wide user base to be threat modeled. Therefore, these interviews serve two purposes: gathering data for this research and doing a preliminary threat modeling for the system. The threat model itself is out of scope of this thesis. When conducting qualitative research, the target group, or interviewees, should be selected purposefully, not randomly (Hirsjärvi et al. 2010, 164). Hence, the interviewees were chosen by their expertise to cover as many domains of the system as possible. Participants and their titles are presented in Appendix 3; however, for the enhanced operational security of the system, the anonymity of the case and interviewees is required. Hence details about the system or the participants are removed from the public version of the thesis.

The questions for the interviews were built with a few guidelines. First, the questions should be neutral and should not imply bias, hypothesis or opinion. Difficult and complicated terms and abbreviations should be avoided in order to prevent misinterpretations. Questions that can be answered either with “yes” or “no” should be refrained from. Finally, each question should support either the research hypothesis or threat modeling. The list of questions asked during the interviews is presented in Appendix 2.

The interview can be divided into the following sections as seen in Appendix 2: introduction, background, threats in general, attackers and adversarial activity, national threats and closing. At the beginning of each interview a short introduction was given. During this section, each interviewee was told about the background and motivation for the interviews as well as provided with general information about the thesis. In addition to asking permission to record the conversation, the lifecycle of the recordings was also described. Lastly, interviewees were encouraged to answer the questions with their own words, since qualitative questions do not have right or wrong answers.

The interviews continued with the semi-structured manner with a set of more open questions related to the background of the interviewees. Some questions had follow-up questions based on the initial answer; hence, the semi-structured approach. These orienting questions were asked in order to align responders and their more in-depth, qualitative answers in context. The purpose of these questions was to provide some background information and a scale on how well each participant knows the system in question as well as how familiar different threat modeling techniques are.

Next, the focus shifted towards threats. The participants were asked to describe the most severe and likely threats against the system including both intentional and accidental threats as well as the lifecycle of the system from planning to end of life. The purpose of these questions was to discover most serious threats to help to focus the more systematic and detailed threat modeling process to the severe and relevant threats, but also to identify if any studied framework addresses all the revealed domains.

With the attacker-centric question pattern, interviewees were then guided towards more intentional and adversarial threats. The participants were asked to name the most serious attacker to confirm that these attackers were listed in the predefined attacker list, which was heavily based on Intel's Threat Agent Library (TAL). Considering the nature of the system under threat modeling, special attention in this section was paid to trusted or privileged users. Moreover, both intentional and accidental perspectives were covered. Next, the list mentioned above were shown to each interviewee and asked which attacker types he/she thinks were possible (Appendix 4). Each threat agent was described with similar words to each participant. The descriptions also relied upon the Intel's TAL. Then, participants were encouraged to bring forward tactics, techniques, and procedures in several different scenarios. In the center of these questions were different objectives that adversaries might be motivated to accomplish, and each interviewee was asked to think how these goals could be achieved.

After these granular questions, a higher level approach was taken with the list of twenty threat scenarios and disruptions disclosed in the National risk assessment as presented in Appendix 5. Since threats related to information and hybrid operations as well as threats against the digital operating environment and cyber domain have now more emphasis on the National risk assessment, the threats from the report were included in the interviews. This also allowed to include more global threats that are not targeted to the system in question per se but can still negatively impact either the system or the users. Each participant was asked if he/she thinks if any of the twenty threat scenarios could impact the availability of the system either directly or indirectly.

During the final phase of closing, only one question was asked. To confirm that the initial list of interviewees was adequate, each participant was asked who they think should be interviewed. Finally, the participants were encouraged to bring forward relevant thoughts that might emerge after the session, the schedule was recapped, and each participant was thanked for participating.

The total duration of the interviews varied from a little less than hour to two and half hours.

5.2 Results

Based on the answers given by the interviewees, every participant had at least the basic knowledge about the system being modeled. As expected, the amount of knowledge varied from deep understanding on technical details, operation or business logic to more vague idea about the users, but also the reasons behind the existence of the system. Each participant also described their specialties and that confirmed the wide enough expertise included in the threat modeling process. Besides security, technical and operational expertise, proficiency from business and customership, jurisprudence, project and risk management, software and service development, security auditing and incident response were included among the interviewees. In addition, client's perspective was also mentioned as a role of one participant.

Some of the participants said that they have a hunch that they know what threat modeling is, while others either had no clear opinion or had a very general intuition what it might be. Based on the answers, threat modeling still seems to be quite unused and unfamiliar method, although half of the interviewees told that they had done some kind of threat modeling previously. When asked to name a methodology or framework, majority responded that they haven't used any specific method, merely informal and ad hoc-type threat modeling, where events and scenarios are drawn and discussed or cards are used to identify threats, was used. STRIDE, DREAD and SWOT were only methods that was named in the answers, but more as existing methods instead of familiar and used methods.

According to the answers, the most severe thing that can happen is the loss of reputation caused by data leak or data breach. Some of the answers indicates very specific types of data breaches or details how sensitive data could leak to unauthorized parties, while other just name the loss of reputation as the worst thing, no matter how it happens. Other severe threats mentioned were a failure to meet the objectives set to the project, loss of clients due to a wrong design decision or quixotic service model, integrity of the data is compromised and the possibility to use the system to infect, attack or gather intelligence from clients, which naturally would once again lead to the loss of reputation.

Most of the usual accidental threats that can have a negative impact to the availability, integrity and confidentiality of the data were mentioned. Problems in the data center, power and network disruptions, hardware failures, accidental but harmful configuration changes, software bugs in the code or in the third-party libraries and components were mentioned as threats that can happen without active and purposeful actions. In addition, threats related to key persons and the slowly decaying security culture in the organization were also brought up.

When asked about the threats that are caused purposefully and intentionally, intelligence and reconnaissance got most mentions. Other threats against the confidentiality and integrity, such as data breach and leak, sabotage and data distortion, were also named. However, when focusing on threats on different phases of the product lifecycle, variation occurred. While intelligence was one of the most remarkable threat during the planning and design phase, more non-technical threats emerged as well. Indistinct roles and responsibilities as well as bad design and poor component choices, lack of resources and time, problems with scalability (also with personnel), sticking with the familiar choices and principles instead of creating something new and efficient and the failure to implement working production model was stated.

Answers related to the most important asset had some variation. While the sensitive information inside the system was brought up frequently, other assets, such as customer processes, contract and agreement details, overall availability, development efforts for new and unique features and relationships between stakeholders and clients were mentioned.

Based on the answers, 92 % of participants named government sponsored or state actor as the most serious or likely adversary. However, when iterating through the possible threat agent list (Appendix 4), other adversaries mentioned by every participant surfaced, as seen in Figure 19. Subcontractors, cyber vandals, untrained and disgruntled employees were considered likely threat agents by all the participants when asked, yet nobody considered them to be as bad as the state actor at first. Surprisingly, apart from thief, information partner and terrorist, all other adversarial archetypes were recognized by at least half of the participants. Notably, the participants were not asked to estimate the likelihood of any threat agent on a scale, merely to estimate if the specific agent is likely to attack the system or not. While the question was formed only to map the potential attackers, the discussion of specific adversary's motives and techniques, as well as examples of scenarios initiated by that adversary, surfaced with almost every participant.

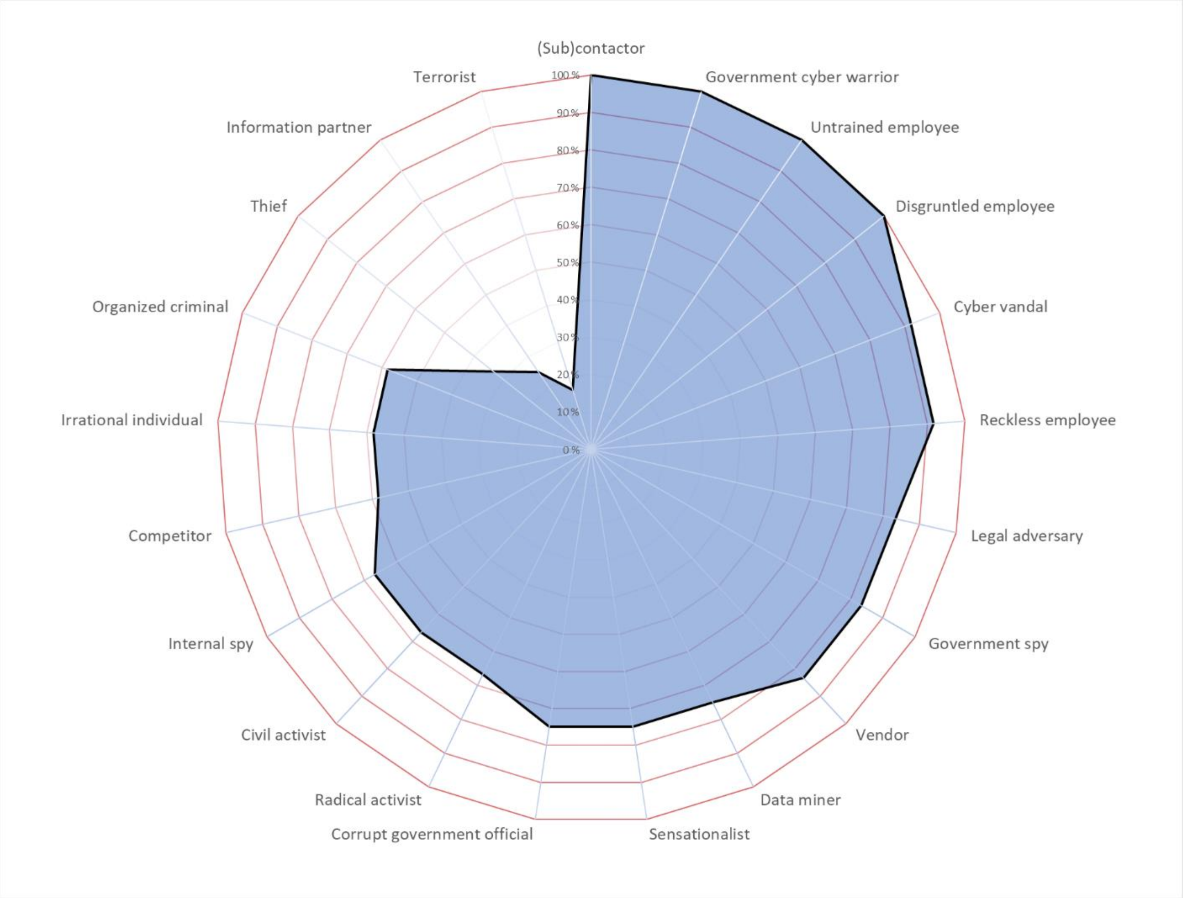


Figure 19. Possible threat agents

The apprehension of threats and disruptions against the society was more homogenous. Flagrant threats, such as disruptions in power supply, fuel availability and communication networks were considered to have an impact to the availability of the service by all the participants. Nine other threat scenarios were taken into account as events that would negatively influence the operation of the system by half of the interviewees. The distribution of answers can be seen in Figure 20.

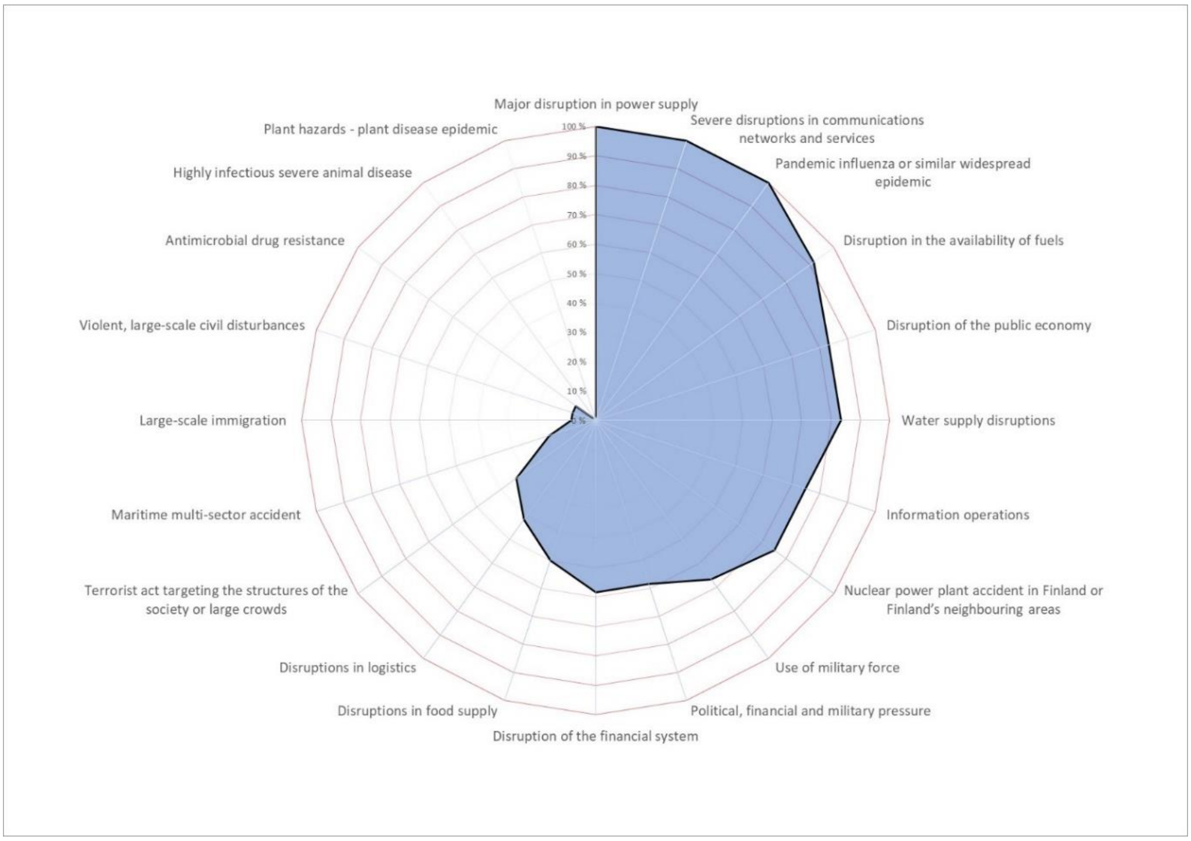


Figure 20. National threats

The answers for the last question about who should be interviewed confirmed that the list of participants was sufficient since no new domains emerged during the responses. Answers mostly emphasized the possibility to have another opinion related to the interviewee's domain.

As a summary, answers reflected multiple dimensions and it would be hard or even impossible to tackle all the surfaced threats with single methodology. For example, intelligence conducted by a foreign nation was mentioned by the majority of the interviewees. Apart from libraries such as ATT&CK, majority of the studied methodologies cover only a portion of the intelligence gathering disciplines, focusing mainly to cyber or digital network intelligence, while other disciplines are mostly disregarded. This is important to note since foreign intelligence agencies predominantly take advantage of open-source, human and signal intelligence and since a foreign nation is the most probable adversarial in this case study, these threats should also be taken into account.

6 Conclusions

6.1 Discussion

The scope of this thesis focuses on threat modeling itself, leaving out mitigation strategies, risk assessment, evaluation, and management. However, it is essential to realize that the threat model can have different purposes and beneficial outcomes at the different levels of an organization. At the system implementation level, threat modeling motivates the building of a more secure system, informs design decisions and security operations and educates staff with more technical aspects and threat intelligence. At the business or mission level, overall enterprise architecture, as well as information security architecture and business function architectures can be scrutinized and motivated to integrate threat intelligence to the processes. An organization’s assumption about its threat environment is an integral part of the overall risk frame, hence threat modeling should reflect and express these aspects as well, and there should be a commitment to the threat modeling at the organizational level as well. Above an organization level threat modeling can provide a common structure for sharing information and threat intelligence, hence supporting interaction between organizations and development of multiparticipant cyber exercises and wargames.

Based on the literature review and answers from a single case study, none of the existing methods or frameworks are suitable for everything and none of these can be recommended over another. This is because threat modeling needs are specific to each project and its requirements. Asset- and attacker-centric approaches can be used to model also non-technical threats, while software and data-centric approaches are designed to be used with software development and operations. However, these can also be applied creatively to other domains as well.

Answers reflected so many non-technical threats, such as project, data protection and information operations related threats, that frameworks focusing on software development or information systems solely, would be insufficient. While loss of reputation was the most severe threat in this use case, techniques to reach that goal varied from technical to psychological threats. Another quite obvious observation was that threat modeling should not focus on issues that other security or safety personnel can quickly and conventionally find by themselves. In other words, threat modeling should focus on issues that other techniques, such as checklists, cannot find. However, it is equally important to identify both relevant and irrelevant threats.

Also, based on the experiences from the case study, threat agents should be included since iterating over list of possible attackers generates useful discussion and guides the direction from accidental threats towards intentional and adversarial threats. Actually, the question about possible threat agents should have been formatted from “is it possible” to “how and why would this agent attack” in order to merge the threat agent list with different scenarios originating from these threat actors. Understanding the attackers, their motivation and skillsets helps to understand also the threats better.

Distinction between threats and risks should have been included in the beginning of each session. Threats are easily mixed with risks and vice versa, and it would be useful if all the participants have a consistent understanding of these terms.

In order to be enterprise-wide, threat modeling requires commitment from all key stakeholders, such as developers, architects, managers and senior executives. A collaborative approach also helps to expose issues related to gaps in the organizations communication channel, gaps in organizations practices and intended effects but also ensures enough diversity in understanding, opinions, and experience which strengthens results of threat modeling. People who use or build the system tends to focus on more technical threats, while people who do not use the system brought up more global threats, such as project-related threats.

6.2 Advices for implementing threat modeling

As the researcher’s opinion, little threat modeling is better than no threat modeling at all, so how could an organization start the threat modeling process without the need to learn all the different methods available? Many of the methodologies share common steps and ideology, so based on the literature review and single case study, following steps can be recommended.

- Decide if the objective is to find all the possible threats or most likely and harmful
- Interview or arrange a workshop with many specialists from many different domains and invite as many stakeholders as possible
- Gather threat intelligence from internal and external sources
- Discover trust boundaries (technical, logical, human) related to the system/service/product in question
- Identify different user groups/types and pay a special attention to privileged users
- Identify most severe threats, both intentional and accidental
- Identify most likely threat agents and their motives and skills
- Identifying most precious assets
- If the goal is to find as many threats as possible, choose and follow a methodology that best serves your needs
- Do threat modeling regularly and start finding threats against mitigations as well
- Document findings, such as threats, threat agents, assets and mitigations
- Ensure that the process is consistent enough in order to get coherent and comparable results between iterations and projects

Choosing a right method can be burdensome and exhausting since much knowledge about different techniques and methods are needed, in order to build a suitable and adequate toolset for different projects. Before selecting a methodology, one should think specific areas, (i.e. privacy, accidental or insider threats), that should be either excluded or included. Additionally, the previous experience with threat modeling and the time and resources available to attach to this process influences the selection as well. Although Shostack (2014) claims that a combination of different approaches (asset-, attackers-, system-centric) tends to be confusing, the complexity of modern systems requires some kind of hybrid approach to the threat modeling. It is important to note, that each method has been developed to address different priorities with different points of view. Therefore, each method has its own strengths and weaknesses and combination of multiple methods is more suitable method than creating a brand new method each time those priorities or needs changes.

Threat modeling should be linked tightly with threat intelligence. While good threat intelligence can give information about the precise actions attackers are currently using, threat modeling focuses on reducing the attack surface in a proactive way. Multiple sources of threat intelligence should be used and each varying source should be correlated to other sources.

Documentation depends on the needs of the project. Suffice to say, some form of documentation should be done. Mind maps are good at the early stages to capture different threats, actors, techniques and their relationships between each other. An example of a threat model in mind map format can be seen in Figure 21.

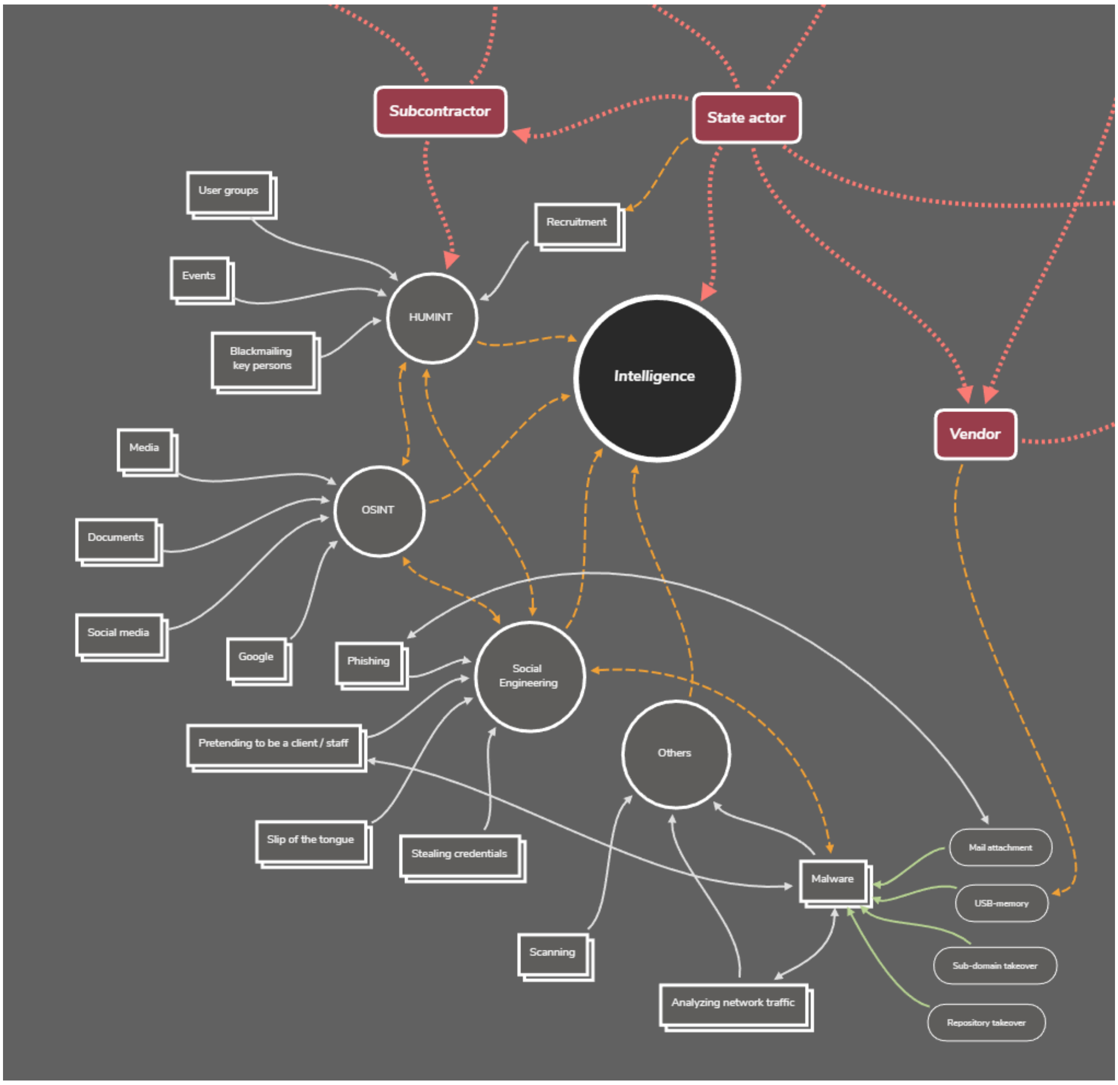


Figure 21. Part of mind map-style threat model

Another suitable and more conservative format is a tabular summary such as threat profile in IDDIL/ATC (Table 5), an attack tree with countermeasures presented in PASTA (Figure 6) or even a forest of basic attack trees (Figure 2). If the goal of the threat modeling is to find every threat, a data flow diagram (Figure 4) should be incorporated, although DFDs from a complex system can quickly become confusing and unmanageable. Presentation of the threat model should be general enough to communicate threats to management and non-technical persons but should also have enough details to satisfy the needs for technical people who are most likely responsible for mitigating the threat.

Threat modeling should also be continuous process where threats against already implemented mitigations are considered as well. Although new threats emerge, the key benefit from regular threat modeling is the accuracy of data resulting from increased frequency in which data is obtained, reviewed and reported.

To better give a guidance which method to choose, all the studied methods and their suitability based on the author's opinion generated from the literature review and the use case are demonstrated in Table 9 below.

Table 9. Suitability of threat modeling techniques based on the results

Methodology	*Suitable alone*	*Suitable when used in conjunction with other methods*	*Suitable for*
Attack trees		X	Documentation and presenting threats
STRIDE		X	Software development
DREAD		X	Rapid risk assessment and analysis
PASTA		X	Software development, organizational risks, business impacts
NIST SP 800-154		X	Data-centric approach
Octave		X	Organizational risks, threats against information assets
CAPEC		X	Understanding and educating adversary behavior
ATT&CK		X	APT TTPs
TARA (Mitre)		X	High-risk adversarial TTPs against cyber assets
TARA (Intel)		X	Threat agents, planning
IDDIL / ATC		X	Reporting, threat intelligence
OWASP		X	Web-based applications (+other platforms)
TRIKE			Automation
VAST			Automation, integration and agile development
Invincea			Exercises
LINDDUN		X	Privacy threats
PnG		X	Threat agents
Security Cards		X	Education,

6.3 Notes about the research

As Yin (2009) noted, certain characteristics can be used in order to evaluate the quality of a case study. Yin describes four characteristics: trustworthiness, credibility, confirmability and data dependability and four tests to be used to evaluate these characteristics: construct validity, internal validity, external validity and reliability. Internal validity should only be a concern when attempting to establish a cause and effect. However, the internal validity has been taken into account with the inclusion of multiple sources (literature, interviews) of data. According to Yin, this is also a good strategy for validating the results of a qualitative research as collecting various types of evidence from different sources makes results more valid. (Yin 2014). Besides using various sources, a researcher can also use a strategy where the results are allowed to be reviewed by the respondents in order to increase the construct validity of a case study; hence, a draft copy of this research was given to the interviewees in order to ensure that the facts and answers have been recorded and transformed correctly.

External validity means that the study and the results are generalizable to other similar situations. A case study should be representative enough to be generalizable. (Yin 2014). The high abstraction layer of the system involved in this case study was mainly due to the operational security; however, it also made the results more representative, thus also more generalizable. External validity could have been increased even further if the validity of findings would had been verified by some external party, or the generalizability had been analyzed more thoroughly by someone else.

A research is considered reliable if another researcher arrives at the same conclusions when following the same procedure. (Yin 2014). Therefore, the process of the literature review has been presented transparently and the sources are listed in the Appendix 1. In addition, the interview questions and the structure of the interview questions are presented in Appendix 2.

6.4 Further development

Since this research relied on single case study, there is an obvious need for verifying the findings in multiple different use cases. Each methodology or framework should also be studied further by applying a specific method on multiple different use cases in order to give a deeper understanding on how thorough a methodology is and thereby evaluate how suitable each methodology is in different scenarios and how well they can be adapted and customized. By contrast, multiple methods should be applied to a single use case to better evaluate the differences, strengths and weaknesses between methodologies.

In addition, the amount of false negatives and false positives per methodology should be studied. False negatives indicate the threats that are missed using a specific methodology, while false positives refer to the threats identified mistakenly, i.e. threats that are not relevant to the subject matter.

One area of suggested further research is platform specific threat modeling. The adoptability of threat models for newish and emerging technologies, trends and phenomena, such as cloud based computing, the Internet of Things, self driving cars, mixed and virtual reality, blockchains, artificial intelligence, quantum computing, should be evaluated and adjusted or new methodologies created if necessary.

References

Bodeau, D. J., McCollum, C. D. & Fox, D. B. 2018. Cyber Threat Modeling : Survey, Assessment, and Representative Framework. McLean: The MITRE Corporation

Caralli, R. A., Stevens, J. F., Young, L. R. & Wilson, W. R. 2007. Introducing OCTAVE Allegro: Improving the Information Security Risk Assessment Process. Pittsburgh: Software Engineering Institute, Carnegie Mellon University.

Casey, T. 2007. Threat Agent Library Helps Identify Information Security Risks.

Cleland-Huang, J. 2014. How Well Do You Know Your Personae Non Gratae? IEEE Software, 31, 4, 28-31.

Deng, M., Wuyts, K., Scandariato, R., Preneel, B. & Joosen, W. 2011. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requirements Engineering, 16, 1, 3-32.

Denning, T., Friedman, B. & Kohno, T. 2013. Security Cards Information Sheet. Accessed on 29 April 2019. Retrieved from https://securitycards.cs.washington.edu/assets/security-cards-information-sheet.pdf

Eng, D. 2017. Integrated Threat Modelling. Oslo: University of Oslo.

Finnish Terminology Centre TSK. 2017. Vocabulary of Comprehensive Security (TSK 50). Helsinki: Finnish Terminology Centre TSK.

Finnish Terminology Centre TSK. 2018. Vocabulary of Cyber Security (TSK 52). Helsinki: Finnish Terminology Centre TSK.

Goodwin, M. 2017. OWASP Threat Dragon. Retrieved April 14, 2019, from https://www.owasp.org/index.php/OWASP_Threat_Dragon

Green, B. N., Johnson, C. D. & Adams, A. 2006. Writing narrative literature reviews for peer-reviewed journals: secrets of the trade. Journal of Chiropractic Medicine, 5, 3, 101-117.

Habib, M., Pathik, B. B. & Maryam, H. 2014. Research methodology - contemporary practices: guidelines for academic researchers. Newcastle upon Tyne: Cambridge Scholars Publishing.

Hirsjärvi, S., Remes, P. & Sajavaara, P. 2010. Tutki ja kirjoita (15.-16. ed.). Hämeenlinna: Kustannusosakeyhtiö Tammi.

Holmberg, J. 2016. Threat Modeling for Train Control and Management Systems based on the Ethernet Train Backbone. Master’s Thesis. Aalto University.

Invincea, Inc.. 2015. Know Your Adversary: An Adversary Model for Mastering Cyber-Defense Strategies. Fairfax.

JAMK (n.d.). Master's Thesis - Studyguide. Accessed on 20 January 2018. Retrieved from https://studyguide.jamk.fi/en/Study-Guide-Masters-Degrees/Studying-at-jamk/masters-thesis/

Kaur, K. & Sharma, R. 2017. Critical: Threat model for an outsourcing business. 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT). Delhi, India.

Khan, R., McLaughlin, K., Laverty, D. & Sezer, S. 2018. STRIDE-based Threat Modeling for Cyber-Physical Systems. 2017 IEEE PES: Innovative Smart Grid Technologies Conference Europe (ISGT-Europe): Proceedings. IEEE.

Launonen, J. 2015. Threat modeling a factory environment using Microsoft Security Development Lifecycle methodology. Master's Thesis: Turun Yliopisto.

Mead, N. R., Shull, F., Vemuru, K. & Villadsen, O. 2018. A Hybrid Threat Modeling Method. Pittsburgh: Carnegie Mellon University.

Mead, N., Shull, F., Spears, J., Heibl, S., Weber, S. & Cleland-Huang, J. 2017. Crowd Sourcing the Creation of Personae Non Gratae for Requirements-Phase Threat Modeling. 2017 IEEE 25th International Requirements Engineering Conference (RE), 412-417. Lisbon.

Miles, M. B., Huberman, A. M. & Saldaña, J. 2014. Qualitative data analysis: a methods sourcebook. Thousand Oaks, Califorinia: SAGE Publications, Inc.

MITRE. 2018. About CAPEC. Accessed on 22 February 2019. Retrieved from http://capec.mitre.org/about/index.html

Muckin, M. & Fitch, S. C. 2015. A Threat-Driven Approach to Cyber Security - Methodologies, Practices and Tools to Enable a Functionally Integrated Cyber Security Organization. Lockheed Martin Corporation.

NIST Special Publication 800-30, Guide for Conducting Risk Assessments. 2012. Gaithersburg: National Institute of Standards and Technology

Pousi, J. 2019. National risk assessment 2018. Helsinki: Ministry of the Interior.

Rosenquist, M. 2009. Prioritizing Information Security Risks with Threat Agent Risk Assessment.

Saitta, P., Larcom, B. & Eddington, M. 2005. Trike v.1 Methodology Document [Draft]. Accessed on 5 January 2018. Retrieved from http://www.octotrike.org/papers/Trike_v1_Methodology_Document-draft.pdf

Scandariato, R., Wuyts, K. & Joosen, W. 2013. A descriptive study of Microsoft's threat modeling technique. Requirements Engineering, 20, 2, 160-180. Springer London.

SFS-ISO/IEC 27005:2011. Information technology. Security techniques. Information security risk management. Finnish Standards Association SFS. 2013.

Shevchenko, N., Chick, T. A., O'Riordan, P., Scanlon, T. P. & Woody, C. 2018. Threat Modeling: A Summary of Available Methods. Pittsburgh: Carnegie Mellon University; Software Engineering Institute.

Shostack, A. 2014. Threat Modeling: Designing for Security. Indianapolis: John Wiley & Sons, Inc.

Sivula, A. 2015. Security Risk and Threat Models for Health Care Product Development Processes. Master's Thesis: Jyväskylä University of Applied Sciences.

Souppaya, M. & Scarfone, K. 2016. Guide to Data-Centric System Threat Modeling (NIST Special Publication 800-154). Gaithersburg: National Institute of Standards and Technology.

Strom, B. E., Applebaum, A., Miller, D. P., Nickels, K. C., Pennington, A. G. & Thomas, C. B. 2018. MITRE ATT&CK™ : Design and Philosophy. McLean. Accessed on 24 February 2019. Retrieved from https://medium.com/mitre-attack/att-ck-101-17074d3bc62

ThreatModeler Software, Inc. 2018. Threat Modeling Methodologies: What is VAST? Retrieved 3 3, 2019, from https://threatmodeler.com/2018/10/09/threat-modeling-methodologies-vast/

Trike. (n.d.). Retrieved January 5, 2019, from http://www.octotrike.org

U.S. Department of Defense 2002. Defense.gov Transcript: DoD News Briefing - Secretary Rumsfeld and Gen. Myers. Retrieved January 20, 2018, from United States Department of Defense (defense.gov): http://archive.defense.gov/Transcripts/Transcript.aspx?TranscriptID=2636

UcedaVélez, T. & Morana, M. M. 2015. Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis. Hoboken: John Wiley & Sons, Inc.

University of Washington. 2009. About. Retrieved April 29, 2019, from Control-Alt-Hack: http://www.controlalthack.com/about.php

University of Washington. 2013. The Cards. Retrieved April 29, 2019, from The Security Cards: http://securitycards.cs.washington.edu/cards.html

Watson, C. & Zaw, T. 2018. OWASP Automated Threat Handbook - Web Applications.

Wynn, J., Whitmore, J., Upton, G., Spriggs, L., McKinnon, D., McInnes, R., Graubart, R. & Clausen, L. 2011. Threat Assessment & Remediation Analysis (TARA). Bedford.

Yin, R. K. 2014. Case study research : design and methods. Fifth edition. Los Angeles: SAGE Publications, Inc.

Appendices

Appendix 1. Summary of sources

Publication	***Source***	***Type***
A descriptive study of microsoft's threat modeling technique	Google Scholar	Journal Article
A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements	Google Scholar	Journal Article
A threat-driven approach to cyber security - methodologies, practices and tools to enable a functionally integrated cyber security organization	Google	White Paper
Critical: threat model for an outsourcing business	IEEE	Conference Proceeding
Crowd sourcing the creation of personae non gratae for requirements-phase threat modeling	IEEE	Conference Proceeding
Cyber threat modeling: survey, assessment, and representative framework	Google Scholar	Technical Paper
Guide to data-centric system threat modeling	Google Scholar	Report
How well do you know your personae non gratae?	IEEE	Journal Article
Integrated threat modelling	Google Scholar	Master's Thesis
Introducing octave allegro: improving the information security risk assessment process	Google Scholar	Technical Report
Know your adversary: an adversary model for mastering cyber-defense strategies	Google	White Paper
Owasp automated threat handbook - web applications	Google	Technical Report
Prioritizing information security risks with threat agent risk assessment	Google	White Paper
Risk centric threat modeling: process for attack simulation and threat analysis	MIT	Book
Security risk and threat models for health care product development processes	Theseus	Master's Thesis
Stride-based threat modeling for cyber-physical systems	IEEE	Conference Proceeding
Threat agent library helps identify information security risks	Google Scholar	White Paper / Technical Report
Threat assessment & remediation analysis (tara)	Google Scholar	Technical Report
Threat modeling a factory environment using microsoft security development lifecycle methodology	Finna	Master's Thesis
Threat modeling: a summary of available methods	Google Scholar	White Paper
Threat modeling: designing for security	MIT	Book

Threat modeling for train control and management systems based on the ethernet train backbone	Google Scholar	Master’s Thesis
Towards a systematic threat modeling approach for cyber-physical systems	IEEE	Conference Paper

Appendix 2. Interview; structure, and questions

Introduction

- Objectives for the interview
- Information about the thesis
    - School
    - Degree Programme
    - Supervisors
    - Topic
    - Schedule
    - Public and supervisor's versions
- Permission to record the conversation and description of how recordings are stored and deleted
- Mentioning that since no customization per interviewee has been done, some question may feel dumb or confusing and there are no right or wrong answers

Background

- How well do you know the system/service/product?
    - Do you know who uses is it, how it is used and/or built?
- How would you describe your strengths and expertize and/or responsibilities and roles in regards to the system/service/product?
- Do you know what threat modeling is?
- Have you previously participated in the threat modeling process?
    - If yes, do you remember what methods or frameworks were used?
        - Did those techniques have any strengths or weaknesses?

Threats in General

- What is the worst thing that can happen to the system/service/product?
- What other threats or harmful situations you think that can occur? Focus on events and scenarios that are not presumably considered by other intervieewes.
- Describe threats that are caused by accident or without active and purposeful actions.
- Describe threats that are caused intentionally or actively.
- What kind of threats do you consider to be relevant to the planning and design phase?
- What kind of threats do you consider to be relevant during the implementation, deployment and maintenance phases of the system/service/product?
- What threats are relevant when the end-of-life of the system/service/product is reached?
- What single component or process should be the most protected thing?

Attackers and adversarial actions

- Who is the most severe or likely attacker or cause of damage?

- What is the worst scenario that following trusted users can cause, either accidentally or purposefully?
    - System administrator
    - Software developer
    - Regular user
    - External auditor
- What following adversarial types do you consider to be possible attackers? (Appendix 4)
- How would an attacker gather information and conduct reconnaissance about the system/service/product, it is users and operation?
- How would an attacker deliver harmful content to the users of the system/service/product?
- What is the best way to would an attacker disrupt the availability of the system/service/product?
- What methods can attackers use to cover or obfuscate the real attack?
- What is the best way to ruin the reputation of the organization or the reputation of the system/service/product?
- Are there any other physical or psychological threats that are worth mentioning?

National threats

- What following threat scenarios and disruptions do you consider relevant, meaning that they might have an impact on the availability to the system/service/product either directly or indirectly? (Appendix 5)

Closing

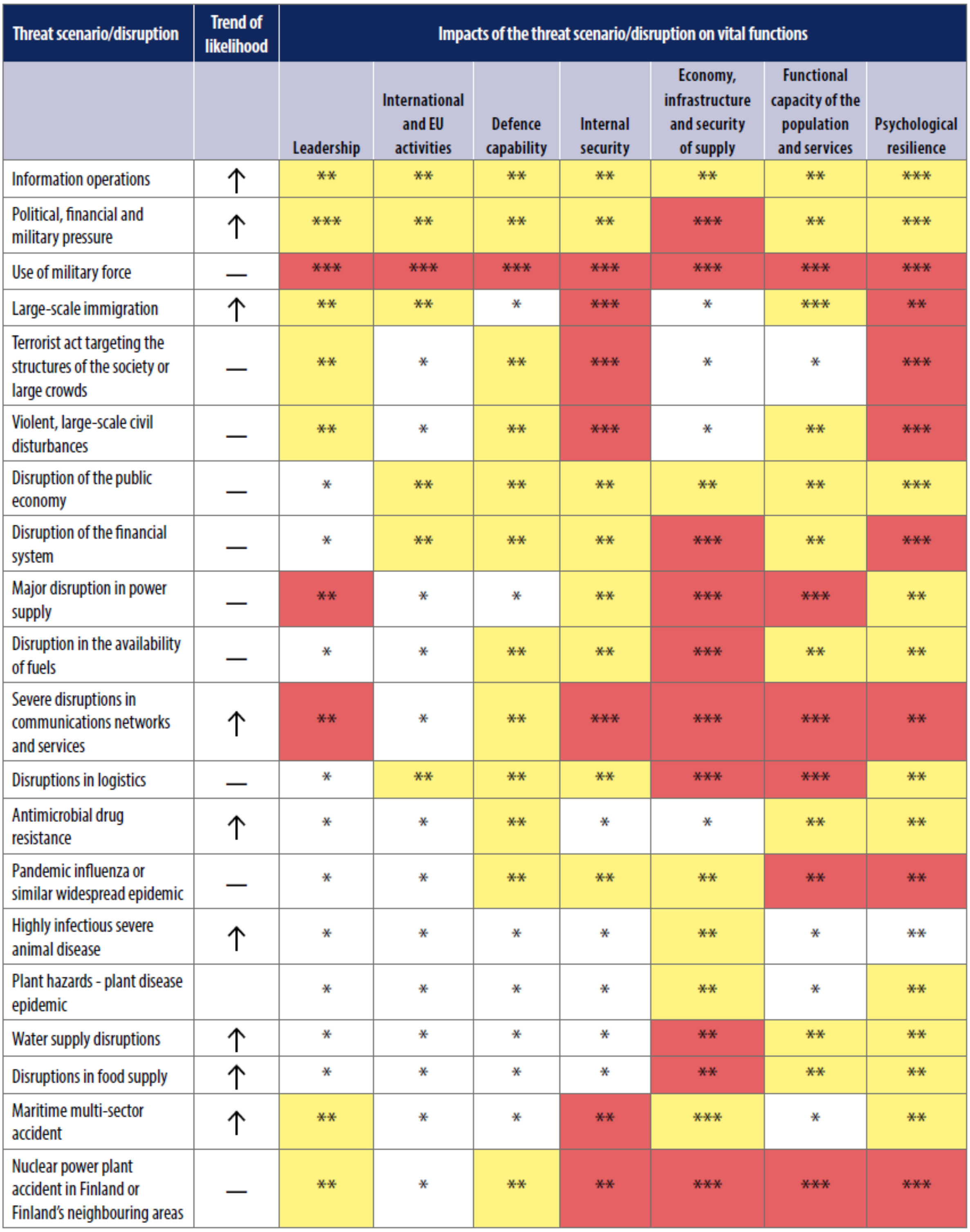
- In your opinion, who should I interview?
- Encourage to report afterward any relevant threats or thoughts surfacing after this session
- Recapping the schedule
- Thanking for the participants

Appendix 3. Interviewees (confidential)

Appendix 4. Attacker list

ENGLISH	SUOMEKSI
(SUB)CONTACTOR	Alihankkija
COMPETITOR	Kilpailija
DATA MINER	Ulkopuolinen datan kerääjä
RADICAL ACTIVIST	Aktivisti, joka on valmis fyysisiin toimiin
CYBER VANDAL	Kybervandaali
SENSATIONALIST	Sensaationhakuinen yksilö
CIVIL ACTIVIST	Aktivisti
TERRORIST	Terroristi
IRRATIONAL INDIVIDUAL	Päättömästi käyttäytyvä yksilö
GOVERNMENT CYBER WARRIOR	Hyvin resursoitu valtiollinen hyökkääjä
ORGANIZED CRIMINAL	Järjestäytynyt rikollisuus
CORRUPT GOVERNMENT OFFICIAL	Virka-asemaansa väärinkäyttävä virkamies
LEGAL ADVERSARY	Lakipahis
INTERNAL SPY	Luotettu henkilö, joka kerää dataa omiin tarkoituksiinsa
GOVERNMENT SPY	Valtion tukema sisäpiiriläinen
THIEF	Varas
VENDOR	Toimittaja (laite, softa, palvelu yms)
RECKLESS EMPLOYEE	Välinpitämätön työntekijä
UNTRAINED EMPLOYEE	Osaamaton työntekijä
INFORMATION PARTNER	Yhteistyökumppani
DISGRUNTLED EMPLOYEE	Tyytymätön työntekijä

Appendix 5. Threat scenarios and disruptions in National risk assessment 2018 (Pousi, 2019, p. 69)



Threat scenario/disruption	Trend of likelihood	Impacts of the threat scenario/disruption on vital functions						
		Leadership	International and EU activities	Defence capability	Internal security	Economy, infrastructure and security of supply	Functional capacity of the population and services	Psychological resilience
Information operations	↑	🟨 **	🟨 **	🟨 **	🟨 **	🟨 **	🟨 **	🟨 ***
Political, financial and military pressure	↑	🟨 ***	🟨 **	🟨 **	🟨 **	🟥 ***	🟨 **	🟨 ***
Use of military force	—	🟥 ***	🟥 ***	🟥 ***	🟥 ***	🟥 ***	🟥 ***	🟥 ***
Large-scale immigration	↑	🟨 **	🟨 **	*	🟥 ***	*	🟨 ***	🟥 **
Terrorist act targeting the structures of the society or large crowds	—	🟨 **	*	🟨 **	🟥 ***	*	*	🟥 ***
Violent, large-scale civil disturbances	—	🟨 **	*	🟨 **	🟥 ***	*	🟨 **	🟥 ***
Disruption of the public economy	—	*	🟨 **	🟨 **	🟨 **	🟨 **	🟨 **	🟨 ***
Disruption of the financial system	—	*	🟨 **	🟨 **	🟨 **	🟥 ***	🟨 **	🟥 ***
Major disruption in power supply	—	🟥 **	*	*	🟨 **	🟥 ***	🟥 ***	🟨 **
Disruption in the availability of fuels	—	*	*	🟨 **	🟨 **	🟥 ***	🟨 **	🟨 **
Severe disruptions in communications networks and services	↑	🟥 **	*	🟨 **	🟥 ***	🟥 ***	🟥 ***	🟥 **
Disruptions in logistics	—	*	🟨 **	🟨 **	🟨 **	🟥 ***	🟥 ***	🟨 **
Antimicrobial drug resistance	↑	*	*	🟨 **	*	*	🟨 **	🟨 **
Pandemic influenza or similar widespread epidemic	—	*	*	🟨 **	🟨 **	🟨 **	🟥 **	🟥 **
Highly infectious severe animal disease	↑	*	*	*	*	🟨 **	*	**
Plant hazards - plant disease epidemic		*	*	*	*	🟨 **	*	🟨 **
Water supply disruptions	↑	*	*	*	*	🟥 **	🟨 **	🟨 **
Disruptions in food supply	↑	*	*	*	*	🟥 **	🟨 **	🟨 **
Maritime multi-sector accident	↑	🟨 **	*	*	🟥 **	🟨 ***	*	🟨 **
Nuclear power plant accident in Finland or Finland's neighbouring areas	—	🟨 **	*	🟨 **	🟥 **	🟥 ***	🟥 ***	🟥 ***