A Framework for Characterizing the Security Posture of Cyber Systems

by

Matthew Ridley

Bachelor of Science
Computer Engineering
Florida Institute of Technology
2017

A thesis
submitted to the College of Engineering and Science
at Florida Institute of Technology
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Computer Engineering

Melbourne, Florida
December, 2018

We the undersigned committee
hereby approve the attached thesis

A Framework for Characterizing the Security Posture of Cyber Systems by

Matthew Ridley

Carlos E. Otero, Ph.D.
Associate Professor
Department of Computer Engineering and Sciences
Committee Chair

Munevver Subasi, Ph.D.
Associate Professor
Department of Mathematical Sciences
Outside Committee Member

Samuel P. Kozaitis, Ph.D.
Professor
Department of Computer Engineering and Sciences
Committee Member

Philip J. Bernhard, Ph.D.
Associate Professor
Department of Computer Engineering and Sciences

ABSTRACT

Title:

A Framework for Characterizing the Security Posture of Cyber Systems

Author:

Matthew Ridley

Major Advisor:

Carlos E. Otero, Ph.D.

Modern day applications can be spread across multiple virtual or physical systems, and be accessed or attacked by pretty much any one any where. Cybersecurity is used to mitigate these cyber threats but there are limited resources that can be dedicated to security. As result, trade-offs and decisions must be made around what is prioritized and what isn't. Cyber risk management provides methodologies for identifying threats, evaluating risks and making decisions, however, it can be difficult to determine whether the system is actually secure enough and the risk is actually within an acceptable parameters. This thesis provides a framework for managing threats and quantifying the security posture, in the form of risk desirability, of cyber systems.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank the amazing people I've worked with in the WICE lab to make this project possible including David Elliott, Xavier Merino, Nicholas Lowing, Evan Martino, and Perter Tarsoly.

I would especially like to thank Dr. Carlos E. Otero who has been an irreplaceable mentor and guide throughout my academic and professional journey. I would not be where I am today without him.

I am grateful for the love an support my family have given me. I thank my mother who's constant love, support, and level headed advice has gotten me through the toughest of times. I thank my late father who gave me the discipline and work ethic to be successful in work and academics. I would also like to thank the rest of family who seem to have no end to the love, support, and guidance they give.

Chapter 1

Introduction

The impact that cyber threats have on businesses is on the rise. According to Kaspersky Labs, the average cost of a cyber breach is $1.23 million [7]. In an environment where the cost risk is so high, having strong cyber security measures in place is as important as ever.

Cyber systems are also becoming larger and more complex. There are large enterprise clouds or unique systems such as the one proposed in [9]. These systems require more advanced methods to ensure security and resiliency such as the method proposed in [6]. However, a business cannot put unlimited money, time, and resources into improving cyber security. They have to set priorities with their limited resources. Because of this, it is important that the business understand risks due to cyber threats that their applications face so that they can allocate resources where they are most useful.

The field of risk management seeks provide methods and tools to allow business to determine risks of threats and allow businesses to make intelligent decisions. However, cyber threats are often complicated multistage events performed by an

intelligent bad actor. Without modeling the lifecycle and various behaviors of attacks, it can be difficult to fully understand the level of risk to the system. Also, the methods in risk management often do not provide a way to evaluate the application as a whole to determine whether the risk is acceptable.

In this thesis, the we provide a framework for managing cyber threats and their risks. This framework includes three parts. First, process for identifying threats to an application. Seconds, a novel method of capturing and modeling threats, and visualizing the risk. Third, a metric use for evaluating the acceptability of the risk of the system.

This thesis is structured as follows:

- In Chapter 2 we go into the background of cyber security, threat and risk management, security metrics. We also present and overview of the literature in the area of cyber risk and threat management.
- In Chapter 3 we describe the various components in the framework. This includes, identifying threats, managing threats and evaluating risk. We also show an example application being ran through the proposed process of threat management.
- In Chapter 4 we conclude the thesis and identify areas of future work.

Chapter 2

Background

The goal of this work is to provide a method for managing cyber threats and risk, and evaluating security posture. This section explains the basic concepts of behind cyber security and risk management. It also goes over the literature for state of the art in risk management.

2.1 Cyber Security and Risk Management

2.1.1 Cyber Security and Threats

Cyber security is the field of protecting cyber assets such as software, hardware and data from threats. There are three areas in which assets are protected by cyber security.

- Confidentiality - Protecting sensitive information from being accessed.
- Integrity - Preventing assets from being modified or destroyed.

- Availability - Ensuring that assets can be accessed by authorized parties.

Threats are anything that can cause harm to an asset. Cyber threats in particular cause harm to software, hardware or data. A common method for categorizing threats was created by Microsoft called STRIDE [11].

- Spoofing - Impersonating someone
- Tampering - Modifying the system or its data
- Repudiation - Disputing who performed a specific action
- Information Disclosure - Accessing sensitive information without proper authorization.
- Denial of Service - Preventing authorized users from fully accessing a resource
- Elevation of Privileges - Gaining privileges without proper authorization

2.1.2 Risk and Risk Management

2.2 Literature Review

The important aspects of this work include modeling threat behaviors, managing threats and risk, and evaluating the overall security and risk of the system. The following is a review of the state of the art literature on these concepts.

2.2.1 Threat Life-cycle and Behavior

For threat management, threats to the system must be identified. However, this can be difficult without first understanding how to represent the life-cycle and

behavior.

2.2.2 Threat Modeling

The purpose of threat modeling is to be able to identify threats to a system. There are many methods of threat modeling proposed in literature.

2.2.2.1 Breaking Down Threat Life-cycle and Behaviors

Understanding threat life-cycle and behaviors can make identifying provides a way of characterizing threats. It also makes it easier to identify threats and determine risk.

An example of a breakdown of threat behavior is provided The Mitre Corporation's Mitre ATT&CK Matrix [16]. This matrix provides a set of threat techniques for a set of attack behaviors. This matrix is meant to be used to help identify threats to enterprise systems. It includes the following behaviors:

- Initial Access - Methods for gaining access to the system. For example, front facing application, spear-phishing, or hardware access.
- Execution - Techniques for running malicious code on the system such as command line interfaces, job scheduling, or user execution.
- Persistence - Methods for having the compromise persist on the system such as saving to files, recurring scheduled jobs, or corrupting kernel modules.
- Privilege Escalation - How the attacker can increase privileges to perform unauthorized actions. For example, injecting into privileged processes, or accessing valid privileged accounts.

- Defense Evasion - Bypassing security measures put in place to protect the system
- Credential Access - Methods of accessing credentials such as passwords, encryption key, or authentication keys.
- Discovery - Methods of discovering users, asset, services or other aspects about the system such as port scanning.
- Lateral Move - Methods of moving a compromise from one system or component to another.
- Collection - Methods for collecting sensitive data.
- Exfiltration - Methods for discretely extracting sensitive information from the system.
- Command and Control - Methods for manipulating or controlling a compromised system such as through an installed remote access tool.

The MITRE ATT&CK Matrix provides a very useful breakdown of threat behaviors but spreads a very wide net. The techniques in this matrix are meant to apply to any enterprise system running a specific operating system. As opposed to the goal of this work which is meant to characterize threats of a specific application. The breakdown also deals with individual behaviors but has no concept of threat life-cycle which is another goal of this work.

LogRythm [13] privides a framework for breaking down cyber threat life-cycle. LogRhythm breaks down threat life-cycle into 6 stages. These steps are:

- Reconnaissance - Analyzing the system and determine a method of attack to compromise the system

- Initial Compromise - Compromise the system.
- Command and Control - Manipulate and control the compromised system.
- Lateral Move - Move between systems until the target has been reached.
- Target Attainment - Target resources has been compromised.
- Exfiltration, Corruption, Disruption - Either export information (confidentiality), modify or delete information (integrity), or interrupt or degrade the service (availability).

This system framework accomplishes the goal of representing activities in terms of a threat life-cycle but there are flaws to this for our purposes. The state of "Target Attainment" does not really capture any sort of behavior. There are also behaviors such as credential access and privilege escalation that are important to capture but don't fit well within any of these steps.

2.2.2.2 Identifying System Threats

Threat modeling often starts with creating some model of the system [12] [8]. Methods of modeling the system include:

- Data Flow Diagram - This is a flow chart of that is used to represent the data flow of an application. [12] uses this model to identify assets and access points in the system so that potential threats can be identified. [8] describes how EMC Corporation uses this model to represent their industry systems in terms of processes, users, and systems, and the data flow between each.
- Network Model - This is a model that represents networked components and the communication between them. Figure 2.1 shows an example of a basic

network model. [12] suggests using this model to represent large networked systems.

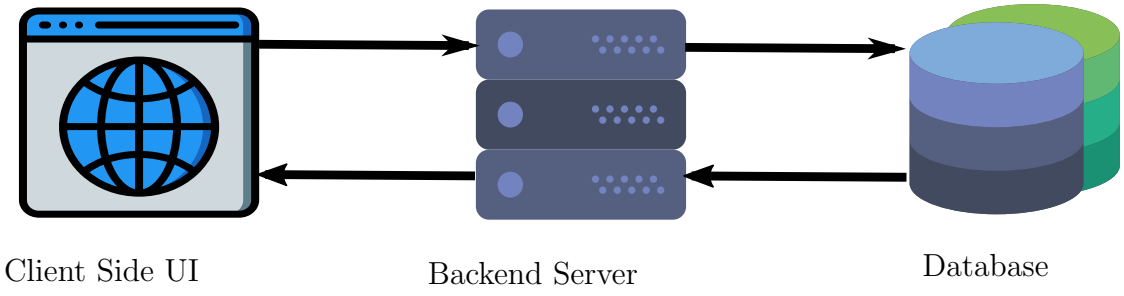

Figure 2.1: Example Network Model

Once a model of the system exists, it can be analyzed to identify threats to the modeled assets.

- The process used in [12] involves identifying attack goals for each system asset in the model, then creating a threat profile by brainstorming for threats keeping in mind threat categories in STRIDE. This process is not very straight forward as it is very much dependent on the expertise and knowledge of the person performing the analysis.

- [8] describes how EMC leverages a threat library to identify threats. This threat library is created for by security experts and was designed for the types of applications that EMC creates. First, interactions between assets in their Data Flow Diagram are analyzed rather than analyzing the assets themselves. The threat library contains the threats that EMC considers worth dealing with as well as information about how applicable the threat would be to a given system. This makes the process straightforward and easy for developers but requires a such a detailed library specific to the type of system being created and maintained.

2.2.3 Sources of Threats

Two of the largest resources for identifying threats to the system are the Common Vulnerabilities and Exposures (CVE) database, and the Common Weakness Enumeration (CWE) database. Both of these databases are maintained by The Mitre Corporation.

The CVE database is a database of commonly known vulnerabilities [4]. It contains descriptions on the vulnerabilities exist in commonly used software, the versions of that software, and the version in which is was patched in.

The CWE database contains a list of commonly known software weaknesses [5]. This deals with general flaws in software with different types of applications, programming languages, and architectures. Unlike CVE it is not concerned with a specific vulnerabilities within specific versions of software [10].

2.2.4 Risk Management

Once threats have been identified for the system, the risk associated with each threat must be determined and managed.

[12] uses several categories for evaluating risk and assigns a value from 1 to 10 for each.

- Number of affected users
- Damage Potential
- Level of skills needed
- Cost of attack
- Reproducibility

- Discoverability

The total risk of the threat is then calculated by the average of each of these categories. One of four things can then be done about the risk:

- Accept the Risk - Cost of mitigating exceeds risk to risk is accepted.
- Transfer the Risk - Risk is transferred to a third party such as insurance.
- Remove the Risk - The component with the risk is removed.
- Mitigate the Risk - A security measure is put in place to mitigate the risk.

In [8], EMC has developers fill out a questionnaire of yes or no questions for each threat which include questions. Based on the answers to this question, a CVSS score is calculated to determine the level of risk. The threats within EMC's threat library are already determined to be worth mitigating and mitigation instructions are included in the library.

2.2.5 Security Posture and Risk Metrics

2.2.5.1 Risk of Individual Threats

There are several methods for determining the risk of threats.

A common way of evaluating risk is with a risk matrix [14]. In this matrix, columns and rows each represent a respective likelihood or occurrence and impact. Each level of impact and likelihood has a specific definition that may vary depending on the context. For example, a e-commerce website may consider risk in terms of downtime, while a government project may consider risk in terms of lives lost or threat to national security.

	Impact				
	Negligible	**Minor**	**Moderate**	**Major**	**Critical**
Almost Certain	Medium	High	High	Extreme	Extreme
Likely	Low	Medium	High	High	Extreme
Possible	Low	Medium	Medium	High	High
Unlikely	Low	Low	Medium	Medium	High
Insignificant	Low	Low	Low	Low	Medium

Table 2.1: Risk Matrix

Another method of evaluating risk is through in the Common Vulnerability Scoring System (CVSS) [2]. This is method often used on CVE database entries to determine a quantified level of risk between 0 and 10 associated with a vulnerability. This value is based on three sub metrics. Base, which is based on things such as attack vector, complexity, and impact, temporal, which is based on the maturity of exploit code, and environmental, which is based on security requirements and importance of the asset being protected. Based on the value of the CVSS score, the severity of the vulnerability is given a qualitative rating of either none, low, medium, high, or critical.

CWSS is method of evaluating software weaknesses created by The Mitre Corporation [3]. It is evaluated in a very similar way to CVSS except it is designed for the software weaknesses held within Mitre Corporation's CWE database.

2.2.5.2 Overall System Security and Risk

There is very little literature on overall system security or risk. This is because it is difficult to fuse different risks together to produce a metric in a meaningful way. The work in [15], quantifies the security posture of system by automatically analyzing containerized systems for known best practices. The existence of a best practice leads to a score of 1 and the absence of that practice leads to a score

of 0 for that feature. The scores of each feature are then fused together using a desirability function to produce a single quantified value for security posture.

There is plenty of skepticism around security metrics in general. [18] presents a survey of literature on quantified security and presents several objections to the state of the art. For one, most quantified security metrics lack empirical testing and comparison to other metrics. Also, there is an inherent uncertainty in any sort of security metric which adds a risk of making incorrect decisions. An inaccurate metric could lead to overconfidence in the security of the system which can be costly for businesses.

These issues are important to keep in mind and are a good reason why single quantified metric while potentially useful, should not be leaned on too heavily. That being said, this work attempts to navigate around risk of overestimating security by creating the metric around risk assessment that is heavily influenced single weaknesses in the system. This is explained further in Section 3.2.4.

Chapter 3

Threat Management

The goal of this chapter is to present the framework of threat management proposed by this work. This framework is built upon the foundations of risk management.

Risk management can be broken down into four steps:

- Identify Risk - Model the system and determine what the risks to assets are
- Analyze and Characterize Risk - Analyze risks to determine individual levels of risk
- Evaluate Risk - Compare, aggregate and group risks determined in the previous step and determine what will be done about each risk
- Mitigate Risk - Plan and perform actions to mitigate risks that are determined to require mitigating.

For the purpose of showing showing how the framework works, an example system was used. This example system is an application called Damn Vulnerable Web Application (DVWA). This is web application composed of a webpage written in

HTML, CSS, and Javascript, server code written in PHP, and a MySQL database. This application is specifically designed to be vulnerable to common threats to web applications such as code injection, cross site scripting, and brute force attacks.

3.1 Identify Threats

This work does not propose one method for identifying threats since there are many ways in which threats can be identified are more suited to specific systems. However we do provide an overview of the process.

The first step to identifying threats is to model the system. As explained in Section 2.2.2.2, this can be done either by a Data Flow Diagram (DFD) or a Network model. For individual applications, a DFD would be more suitable while a Network model would be more applicable to a networked system. Figure 3.1 shows a network model showing the components and communication that take place through the DVWA application.

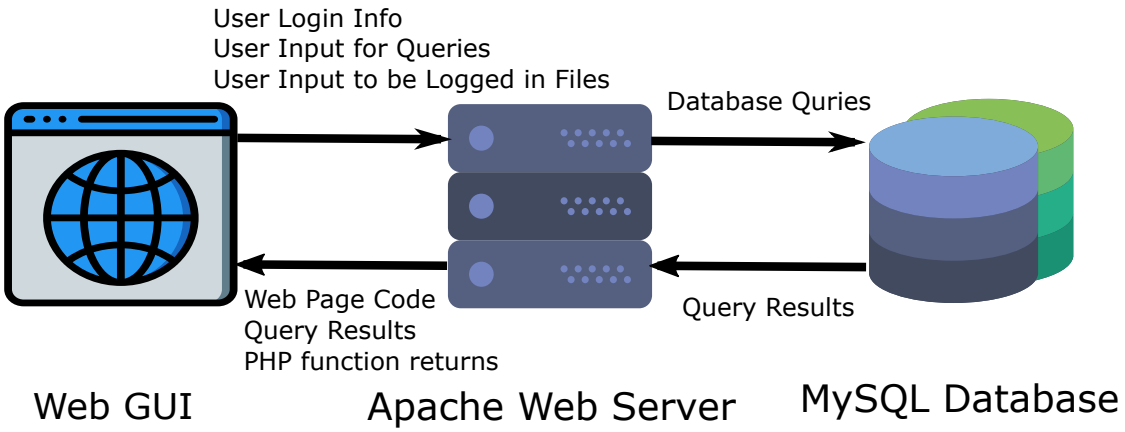

Figure 3.1: DVWA Network Model

Once a model is created threat must be identified. There is no straightforward way to do this. However, review of the literature has not shown any straight

forward way of doing this. First, we start by going through each asset and communication in the model. Threats can then be determined through a combination of brainstorming using STRIDE [12], using databases such as CVE or CWE to determine software vulnerabilities and weaknesses, or using automated static analysis tools.

3.2 Analyze and Characterize Threats

3.2.1 Threat Breakdown

In order to be able to model threats and attacks, the framework breaks down the techniques used in an actual attack. This breakdown combines the positives of both LogRhythm's threat life-cycle breakdown [13] and the MITRE ATT&CK Matrix [16] and classifies behaviors into 6 categories. These categories are Initial Access, Execution, Persistence, Credential Access, Privilege Escalation, Lateral Move, and Exfiltration, Corruption, and Disruption.

3.2.1.1 Initial Access

The Initial Access category deals with how the attacker is able to access the system. This can include, a web services that the public can access, connecting through open ports, or even physical access to the system. Protecting initial points of access can prevent attackers from being able to attempt an attack in the first place.

3.2.1.2 Initial Compromise

The Execution category deals with how the attacker is able to exploit the system on order to compromise it. For example, the attacker perform and SQL injection

attack to compromise a database.

3.2.1.3 Command and Control

The Command and Control category deals with how the attacker is able to control and manipulate the system. This can involve techniques installing remote access tools. This is not a necessary point in the threat life-cycle but an attacker may choose this route if they want to control the actions of the system for a bot-net for example.

3.2.1.4 Credential Access

Often times, access to a system, specific resources, or information on the system will require some sort of credentials such as a username and password, or an encryption key. Methods in the Credential Access category deal with the ways that attackers gain access to these credentials. Examples of this include, brute force or dictionary attacks, looking through code or credential files, phishing or looking through command or bash histories. This is an optional step in the life-cycle that can occur at any step as necessary for the attacker.

3.2.1.5 Lateral Move

Once an attacker has compromised one system, application, or service, they may attempt to compromise other applications or systems that are nearby or on the network. Techniques for moving a compromise from one application/system to another are part of the Lateral Move category. Examples of this include SSH hijacking or exploiting services or communication used by the two systems. This step is performed when the attacker is after some asset outside of the current area

Initial Access	**Initial Compromise**	**Command and Control**	**Credential Access**	**Lateral Move**	**Exfiltration, Corruption, Disruption**
Web UI	SQL Injection	Install Remote Access Tool	Spearphishing		DDOS
Used Open Ports	HTTP Injection	SSH	Read Files on System		HTTP Response with Unauthorized Data
Unused Open Ports	PHP Injection		Brute Force		Modify or Delete Database Data
	File Injection		Sniffing on Connections		

Table 3.1: Initial Threat Management Matrix

of compromise.

3.2.1.6 Exfiltration, Corruption and Disruption

After the attacker has compromised the asset it is after, the attacker can cause harm to confidentiality, integrity, and/or availability. Examples of this include reading sensitive file data, modifying data in databases, or DDOS attacks. Once the compromise of the targeted asset is performed, the attacker's mission will be completed once it has performed something within this step in the life-cycle.

3.2.2 Matrix

Similarly to the MITRE ATT&CK Matrix [17], the framework uses a matrix to capture and organize the techniques into their respective categories. Each threat behavior category is represented by a column in the matrix and each individual threat behavior is captured in each cell.

This matrix provides a document of the security assumptions about what threats exist for the system.

Initial Access	Initial Compromise	Command and Control	Credential Access	Lateral Move	Exfiltration, Corruption, Disruption
Web UI	SQL Injection	Install Remote Access Tool	Spearphishing		DDOS
Used Open Ports	HTTP Injection	SSH	Read Files on System		HTTP Response with Unauthorized Data
Unused Open Ports	PHP Injection		Brute Force		Modify or Delete Database Data
	File Injection		Sniffing on Connections		

Table 3.2: Threat Management Matrix with Risk Visualization for DVWA

3.2.3 Characterizing and Analyzing Threat Risk

Now that threats have been captured in the matrix, they can be analyzed to determine their individual threat level. We can use the risk matrix as explained in Section 2.2.5.1 to determine the risk level of each element in the matrix.

By determining the level of risk, we can now visualize risk in the matrix to get an overview of what tasks to prioritize as well as a rough idea of how much risk is in our system based on the color of each individual cell. Note that the level of risk for "Initial Access" is determined by whether or not the method is an intended way of accessing the system as intended methods of access, this is because accessing the system does not have risk in and of itself unless it is done in an unauthorized way.

3.2.4 Evaluating Risk and Acceptability

The risks have been characterized and given qualitative values, we would like to now fuse aggregate and fuse this information so that we can quantify individual risks, as well as create a metric based on how acceptable the total system risk is.

In this problem we have multiple factors (threats of each behavior type) that we want to fuse into a single value that we want to optimize (the total acceptability

of this risk). Such is called a simultaneous optimization problem.

3.2.4.1 Desirability Function

Similar to the work in [15], this work uses a desirability function to fuse risk together into a single metric. In this problem and function there are multiple factors each influenced by multiple features that must be combined to compute a single value or metric. Each factor is quantified as a response y_i. These factors are transformed into desirability values d_i based on the following equations.

$$d_i = \begin{cases} 0 & y_i \geq U \\ (\frac{y_i - U}{T - U})^r & T \leq y_i \leq U \\ 1 & y_i < T \end{cases}$$

This results desirability values where $0 \leq d_i \leq 1$. In this equation L is the lower bound or the lowest acceptable value for factor y_i below which the desirability is 0. T is the target value above which the desirability is 1. The value r is a weight factor.

Next we want to evaluate each factor y_i For the framework we have a matrix of several threat technique categories each with several techniques within them. For this example each of the matrix categories would indicate a factor and the presence of a mitigation or response to a threat technique would a feature. For a set of m threat techniques in a category, a score of 1 is given to the feature if a mitigation is present and a score of 0 is given if a mitigation is absent. For each feature there is also a weight factor w_j to represent the importance of that feature. The score and weight vectors are shown below.

$s = [s_1, s_2, ...s_m]$ $w = [w_1, w_2, ...w_m]$

The factor security response can then be calculated from the dot product of

		Impact				
		Negligible	**Minor**	**Moderate**	**Major**	**Critical**
		1	**2**	**3**	**4**	**5**
Almost Certain	**5**	5	10	15	20	25
Likely	**4**	4	8	12	16	20
Possible	**3**	3	6	9	12	15
Unlikely	**2**	2	4	6	8	10
Insignificant	**1**	1	2	3	4	5

Table 3.3: Quantified Risk Matrix

these vectors.

$y_i = sw^\top$

These factor security responses can then be plugged into the equation for calculating desirability d_i. A combined single desirability value D can then be calculated from the geometric mean of the individual factor disabilities.

$D = (\prod_{i=0}^{m} d_i)^{\frac{1}{m}}$

For this framework the score vector s is represented by the risk of each threat. To keep things simple all weights in these equations are set to 1.

All that is left is determining the score values from the risk. There is more than one way to get a numeric value for risk. For example, using CVSS scores. To keep things simple we apply numeric values to the risk matrix as shown in Table 3.3.

These values in this updated risk matrix can then be applied to the threat management matrix. The desirability of the risk can then be calculated for each factor and overall. Table **??** shows the example matrix with risk values quantified the desirability calculated. The upper limit for the desirability function was set to 8* the number of rows and the target value was set to 4 * the number of rows in each column. The nature of the metric is that if one row comes out to 0% desirability,

Initial Access 100%	**Initial Compromise 0%**	**Command and Control 25%**	**Credential Access 0%**	**Lateral Move**	**Exfiltration, Corruption, Disruption 0%**
Web UI 1	SQL Injection 25	Install Remote Access Tool 10	Spearphishing 8		DDOS 15
Used Open Ports 1	HTTP Injection 16	SSH 4	Read Files on System 12		HTTP Response with Unauthorized Data 20
Unused Open Ports 6	PHP Injection 20		Brute Force 15		Modify or Delete Database Data 25
	File Injection 20		Sniffing on Connections 10		

Table 3.4: Threat Management Matrix with Quantified Risks. Desirability = 0%

the overall desirability is 0%. As a result any weak link in the chain can have a massive affect on the metric somewhat mitigating the issue of overestimation of security.

3.2.5 Mitigating Threats

The goal in mitigating threats is to lower the risk so that we can improve the desirability of the application. Based on looking at the current threat management matrix the biggest concerns are the various forms of code injection, attacks on the database, and the server returning sensitive data.

The following mitigation techniques were put in place:

- Used parameterized queries to prevent SQL injection attacks
- All data returned by the server is put through a filter to filter out sensitive data
- Server sanitizes and strips any unused HTTP header data
- The input going into PHP functions is sanitized

- Captchas were added to prevent brute force attacks login attacks
- Unused ports were closed
- All connection were encrypted

These specific mitigation decisions were made based on impact to risk compared to cost of mitigating. For example, DDOS was not mitigated despite being a HIGH risk because having the extra server power to handle DDOS attacks is very costly. The mitigation techniques each significantly reduce the likelihood that the threat they are mitigating would occur.

As a result of these mitigation techniques, many of the risk in the threat management matrix have been reduced. This new matrix is represented in Table 3.5. The matrix, visualizes the change in risk. Just by looking at this matrix one can intuitively know that they have a more desirable system. We also now no longer have any categories of threat behavior that have 0% desirability meaning that we get a desirability value. In this case we get desirability of 76.8% overall.

The methods for mitigation all decreased the likelihood of threats. The risks of these threats could be lowered even further by reducing impact. For example,

Initial Access 100%	**Initial Compromise 93.8%**	**Command and Control 87.5%**	**Credential Access 81.25%**	**Lateral Move**	**Exfiltration, Corruption, Disruption 40%**
Web UI 1	SQL Injection 5	Install Remote Access Tool 5	Spearphishing 8		DDOS 15
Used Open Ports 1	HTTP Injection 4	SSH 4	Read Files on System 4		HTTP Response with Unauthorized Data 4
Unused Open Ports 1	PHP Injection 4		Brute Force 3		Modify or Delete Database Data 5
	File Injection 4		Sniffing on Connections 4		

Table 3.5: Threat Management Matrix with Mitigations. Desirability = 76.8%

the impact of data being modified in the database could be decreased by keeping regular backups of the database.

Another thing to mention is that many threat behavior impacts and mitigation techniques overlap. For example, SQL injection is so dangerous because it can cause data to be modified or leaked from the database. Putting in mitigations to SQL injection reduces the likelihood of both database modification and SQL injection.

Chapter 4

Conclusion and Future Work

4.1 Conclusion

Cyber security is an important field that protects systems from cyber threats. It is important to understand these threats and the risk they pose to the system. This thesis analyzed the literature on cyber threat management and described a framework for managing risk from cyber threats and quantifying the risk acceptance of an application.

We accomplished threat management by first identifying threats. This was done by first creating a model of the system assets and its communications, then going through each and identifying threats through research, automatic tools and brainstorming.

Second, we managed and analyzed the threats created using a novel threat management matrix to organize threats by life-cycle behavior and visualize risk based on results of a risk matrix.

Third, we evaluated the risk of threats and presented a novel method for quan-

tifying the risk acceptability of the entire system using a desirability function.

Lastly, we prioritized which threats to mitigate and updated the threat management matrix to reflect the improvements to the desirability of the system.

4.2 Future Work

One of the goals of this thesis for breaking down threat behavior was to ensure that it captured the life-cycle of threats. However, this is not taken advantage of much in this work. Future work could involve using this threat life-cycle model to detect and model attacks in real-time in similar way to the work done by Stack Rox [1].

Another area of future work would be to come up with a method for making threat identification more straightforward and streamlined so that it doesn't require as much time, effort, and knowledge. One possibility would be to automatically generate the threat management matrix based on automatic analysis of the system.

One major limitations of this metric is it does not take into account the cost of mitigating risk. This is a very important part of the process of decision making when it comes to implementing security measures and more work could be done to either fuse it into the metric or make it part of the risk assessment process.

Bibliography

[1] Container attack and runtime detection stackrox: Container security for docker and kubernetes. Available: https://www.stackrox.com/runtime-detection/.

[2] Cvss v3.0 specification document. Available: https://www.first.org/cvss/specification-document.

[3] Common weakness scoring system, Sep 2014. Available: https://cwe.mitre.org/cwss/cwss_v1.0.1.html.

[4] About cve, Nov 2018. Available: https://cve.mitre.org/about/index.html.

[5] Common weakness enumeration, Mar 2018. Available: https://cwe.mitre.org/about/index.html.

[6] Xavier Merino Aguilera, Carlos Otero, Matthew Ridley, and David Elliott. Managed containers: A framework for resilient containerized mission critical systems. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pages 946–949. IEEE, 2018.

[7] Denise Berard. Kaspersky lab report: The cost of a data breach continues to grow worldwide, May 2018. Available: https://usa.kaspersky.com/about/press-releases/2018_kaspersky-lab-report-the-cost-of-a-data-breach-continues-to-grow-worldwide.

[8] Danny Dhillon. Developer-driven threat modeling: Lessons learned in the trenches. *IEEE Security & Privacy*, 9(4):41–47, 2011.

[9] David Elliott, Carlos Otero, Matthew Ridley, and Xavier Merino. A cloud-agnostic container orchestrator for improving interoperability. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pages 958–961. IEEE, 2018.

[10] Daniel Miessler. The difference between cwe and cve, Oct 2017. Available: https://danielmiessler.com/blog/difference-cve-cwe/.

[11] Michael Muckin and Scott C Fitch. A threat-driven approach to cyber security: Methodologies, practices and tools to enable a functionally integrated cyber security organization. *Lockheed Martin Corporation, Available: http://lockheedmartin. com/content/dam/lockheed/data/isgs/documents/Threat-Driven% 20Approach% 20whitepaper. pdf, Accessed*, 12(07), 2017.

[12] Suvda Myagmar, Adam J Lee, and William Yurcik. Threat modeling as a basis for security requirements. In *Symposium on requirements engineering for information security (SREIS)*, volume 2005, pages 1–8. Citeseer, 2005.

[13] Chris Peterson. The threat lifecycle management framework, 2017.

[14] Atle Refsdal, Bjørnar Solhaug, and Ketil Stølen. Cyber-risk management. In *Cyber-Risk Management*, pages 33–47. Springer, 2015.

[15] Matthew Ridley, Carlos Otero, David Elliott, and Xavier Merino. Quantifying the security posture of containerized mission critical systems. In *SoutheastCon 2018*, pages 1–4. IEEE, 2018.

[16] Blake E Strom, Joseph A Battaglia, Michael S Kemmerer, William Kupersanin, Douglas P Miller, Craig Wampler, Sean M Whitley, and Ross D Wolf. Finding cyber threats with att&ck-based analytics. 2017.

[17] Blake E Strom, Joseph A Battaglia, Michael S Kemmerer, William Kupersanin, Douglas P Miller, Craig Wampler, Sean M Whitley, and Ross D Wolf. Finding cyber threats with att&ck-based analytics. 2017.

[18] Vilhelm Verendel. Quantified security is a weak hypothesis: a critical survey of results and assumptions. In *Proceedings of the 2009 workshop on New security paradigms workshop*, pages 37–50. ACM, 2009.