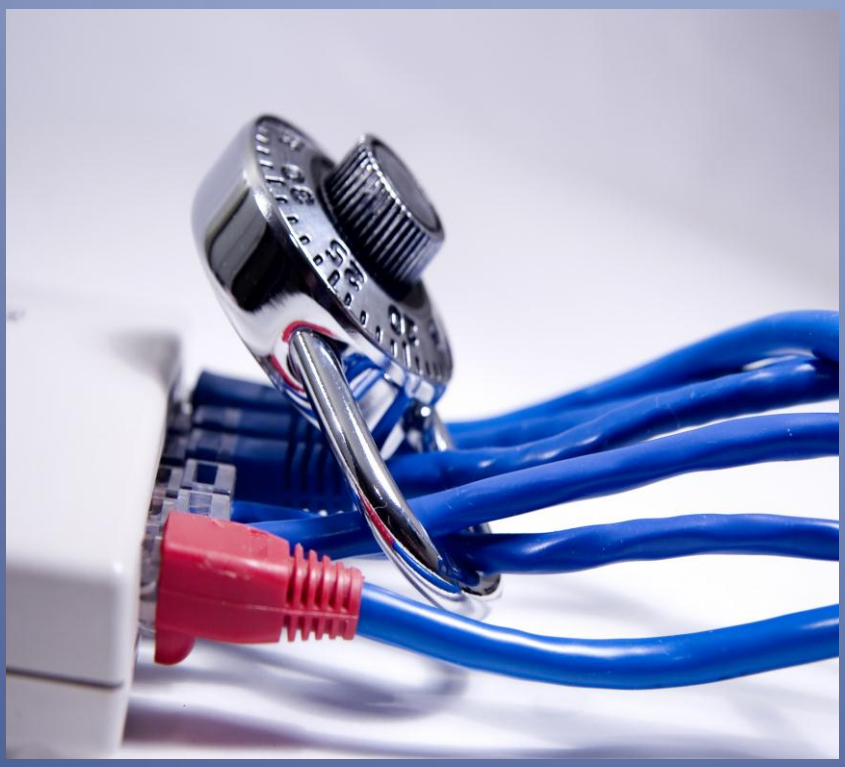

A proactive approach to security integration in web applications

Literature review

Sanele Macanda

Supervisor: Dr. Anne Kayem

Table of Contents

1. Abstract

Web applications have increased rapidly in popularity in the past few years [15]. Application security in web applications like social networks has become a major concern in recent years and so, security has become a strong requirement for all web applications. Therefore, it is important to analyze and model potential threats in a system before implementation [3]. In this literature review we discuss some threat modeling approaches in relation to our project, highlight the pros and cons of each. We explain why threat modeling is essential to enforcing web application security. We look at the attacks and vulnerabilities that are typically found in social networks.

2. Introduction

The growing number of internet users attests to the fact that the world is becoming increasingly technology driven [15].We have chosen to focus on social networks because they are an example of a web application that involves multiple user interaction and raises a lot of issues that are centered on securing access to shared data. Web applications are quite popular and environments such as social networks that encourage data sharing and seamless user interaction facilitate the use of internet. This makes them one of the prime targets to attack because these web applications typically handle considerable amount of personal information. Web applications can be exposed to different types of attacks which weaken the security of the application which makes the information the application manipulates vulnerable. This is a serious problem because organizations that rely on these web applications for business interactions can lose valuable data. Losses of information can damage the reputation of the organization and often costly to repair.

This literature synthesis is centered on using threat models to evaluate web applications in order to discover vulnerability sources before the application is deployed.

3. Threat Modeling

3.1. What is a threat

A threat is an undesirable event that may be malicious [12]. Threats can be exploited to cause serious damage to a system or application and the information manipulated by such systems.

3.2. What is Threat Modeling

Threat modeling is a security control performed during the construction and design phase of the Secure Development Life Cycle (SDLC) to identify and reduce risk within software [4]. The advantage in this approach is that the vulnerabilities can be handled before the application is deployed and in this way it provides stronger information security to users.

3.3. WHY use Threat modeling

A threat modeling is a tool typically used to proactively identify threats that can be classified as potential vulnerabilities in a system and provide countermeasures to prevent these attacks from being exploited to cause damage to the system [1]. Threat model can be used at the end of the design cycle but this can be sometimes make applying security patches a cumbersome task [3]. Threat modeling helps shape your application design to meet security objectives [16].

4. Threat modeling process

The threat modeling is an interactive process that is implemented from the design phase of the application to the end of the application life cycle [19]. This process cannot be done is a single pass as it cannot guarantee that it will identify all threats in an application. Since applications are rarely static and need to be enhanced to meet new business requirements, threat modeling process should be repeated as the application evolves. Figure1. Below shows the threat modeling process

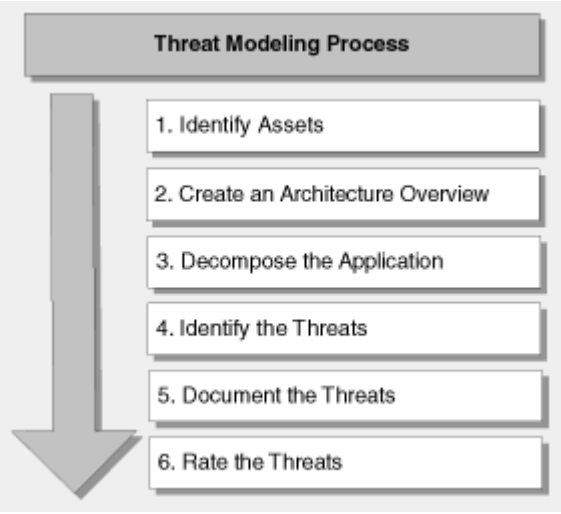

Figure 1. Threat Modeling Process

4.1. Identify Assets

In this step you identify assets, i.e. confidential data of all users in the database of the social network, availability of data, and web server computer.

4.2. Create an architecture overview

In this step you create a high level structure of the application showing the physical deployment components.

4.3. Decompose the application

In this step, the application is broken down to create security profile for the application on typical area where it vulnerable, for example authentication, data flow, entry points and privilege code.

4.4. Identify Threats

In this step, you identify threats that can affect your application assets. Threats can be identified using the STRIDE model, h**o**w this model works will be explained under the Microsoft STRIDE model.

4.5. Document the Threats

In this step, you document the threat by; the type description, where the threat is targeting, attack techniques and countermeasure that could be taken to mitigate the threat. The table below shows the how this is done.

Threat Description	Attacker obtains authentication credentials
Threat target	Web application user authentication process
Risk	Low
Attack techniques	Use network monitoring software
countermeasures	Use SSL to provide encrypted channel

4.6. Rate Threats

In this step, threats are being rated according to their risk. For example if a threat has a high risk then it must be mitigated first.

5. Types of threat modeling approaches:

- **Attack centric**: views the system in an attack perspective. In this approach you look at your system and try to see where it is vulnerable.
- **Defense centric**: evaluates weaknesses in security controls. This approach looks to strengthen these weakness found on the system.

- **Asset centric**: evaluates assets classification and value. This approach looks at the assets on the system and classifies them according to their importance. High classification means higher security measures on the asset.
- **Hybrid centric**: is evaluates application design using methods to meet security objectives.

6. Microsoft STRIDE MODEL

The Microsoft threat model is defense centric, and it uses the STRIDE model to categorize threats [14] which is an acronym spoofing, tempering, Repudiation, Information Disclosure, Denial of service and Elevation of service. This is important because each category has a specific set of mitigations; once threats are analyzed and categorized you can mitigate them [17]. The table below shows each threat to its corresponding category.

Threat	**Security property**
Spoofing	Authentication
Tempering	Integrity
Repudiation	Non-repudiation
Information Disclosure	Confidential
Denial of service	Availability
Elevation of service	Authorization

- **Spoofing** - occurs when an attacker impersonate someone they not by trying to use the user name and password. This type of threat typically happens at the authentication of a web application.
- **Tempering with data** – occurs when an attacker modify data maliciously in order to hamper the system. The attacker would be violating data integrity security property.
- **Repudiation** – occurs when an attacker performs an action and claims that they did not do it
- **Information disclosure** – occurs when an attacker gains permission without a valid authentication. This type of attack will be violating the confidentiality security property.

- **Denial of service** – occurs when an attacker deny service to valid users. These attacks violate the availability in security property. You can stop these attacks by removing the resources that the attacker used, which means valid users can't access the resources either [17].
- **Elevation of service** – occurs when an unprivileged user gains access to a system. These types of attacks violate the authorization security property.

7. Attack trees against STRIDE model

Attack trees model threats in a tree structure, where the root node is the main target and the children being various ways of achieving that goal. Attack trees are used to define and analyze possible attacks on a system in a structure way [20]. The Figure below represents an attack tree. The stride model categorizes threats and each threat can damage the targeted area, each threat has its own target. Whilst the attack tree threats are organized in a tree structure and there is only one target which is the root node. It is easier to circumvent from the attack trees and opposed to the STRIDE model. In the attack tree there is only one target and some paths to the root are not possible. Whilst in the STRIDE model threats have different targets and you must circumvent attacks for all targets to secure the application.

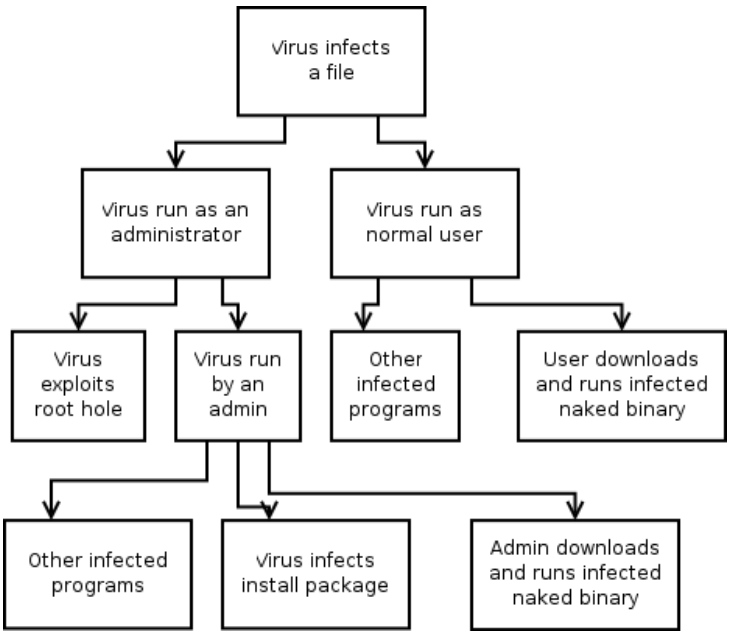

Figure2. Attack tree

8. Type of attacks on web application

8.1. SQL injection attacks

Open Web Application Security Project (OWSP) has released a list of the top ten most common vulnerabilities in web applications [6]. SQL injections are ranked 1st on the list which shows they can be a serious concern in applications that uses a database. Databases are used in many web applications and so it makes sense to take measures to enforce protection of the data.

Threat modeling can be used to identify threats like the SQL injection attacks. This type of an attack focuses on the database of a system. The security threat this attack poses on a web based application would be the built-in database access. SQL injection attacks are caused by attackers who insert a malicious SQL query into the web application to manipulate data or even gain access to the back end of the database. This can be prevented if developers model the application which would give them a better understanding of the system and help identify bugs in the system.

8.2. Shielding against SQL injection attacks

SQL injection attacks can be prevented by using ADMIRE model which is a systematic approach, where you first Analyze the security objectives, Divide the application, Mark vulnerabilities in the system, Identify threats, Rank threats and then Eliminate threats [7].

- **Analyze the security objectives –** security objectives are goals and constrains related to confidentiality, availability and integrity of data**.** Data is the main asset to protect in a database. Data protection must satisfy integrity of the stored data and its confidentiality in a database. Also data should always be available for authorized users in a database.
- **Divide the application** – An application can be divided by means of Data Flow Diagram (DFD) analysis. DFD's is a high-level way of focusing on data and how it flows through the application [7]. The data needs to be analyzed for example when looking at authentication it is important to analyze how data enters the application, how it is processed and how it is stored. SQL injections attacks typically occur at the entry point and it is very difficult to control the entry point because it is very difficult to make assumptions about a user's identity.

- **Mark vulnerabilities in a system** – Marking vulnerabilities can be done by using the STRIDE model. This will categorize these vulnerabilities which makes it easy to mitigate them.
- **Identify threats** - identifying threats can be easy using threat trees. Once a threat has been found using threat trees, the correlated target will also be identified therefore revealing the attacker's intent.
- **Rank threats** – Once these threats have been identified using threat trees, and then the DREAD model is used to rank them form high to low risk. Threats with a high risk will be mitigated first.
- **Eliminate the threat** – High risk threats will be reduces to mitigation actions. Threats will only be eliminated if the application must be using a secured coding practice.

8.3. LDAP and Blind LDAP injections attacks

Other attacks are Lightweight Directory Access Protocol (LDAP) injection techniques. These types of attacks are similar to SQL injection techniques, in both techniques attacks are targeted to the database. LDAP is input validation related, so this technique takes advantage of parameters introduced by the user to generate LDAP query [14]. If LDAP vulnerabilities are not identified and addressed, a malicious user could inject code that is harmful to the web application in the parameters transmitted.

Web applications are vulnerable to Blind LDAP injections that occur when an attacker makes inferences about the nature of the data from a server response. In this case the application will not show an error message from the code injected by LDAP. The filter will generate a valid response (true result or false result). According to Chema [14] the LDAP and Blind LDAP injection techniques can be prevented if defensive programming, sophisticated input validation, dynamic checks and static source code analysis techniques are used.

It is also important that to use threat models to prevent other attacks such as scripting. For instance Ron Bowes wrote a script code that collected personal information of several users on a web based application (Facebook) [9]. If the system is vulnerable to injection attacks then the privacy of the personal information of the users of the system could be compromised. Using a threat modeling tool to identity vulnerabilities that can be exploited to provoke attacks such as injection attacks can go a long way towards enhancing the security of a web application.

9. Information Flow

Other methods of analyzing information flow control include information flow control which essentially requires the programmer to verify the code of the entire system to ensure that information is shared in ways that satisfy the constraints of the security policies. Information flow control is an important consideration when using threat models to secure a system because; oftentimes attacks are centered on weak implementations. Myers [9] pioneered work in the area of information flow control, focusing specifically on statically checking programs to ensure that the information flow protects and ensure integrity of sensitive data according to the specifications of an access control model

10. Vulnerabilities in Social Networks

10.1. SQL injection attacks

As mentioned before our focus is on social networks because they are vulnerable to SQL injection attacks. Social networks uses databases to store user data, SQL injection attacks focus on database as mentioned before.

10.2. Spoofing

Social networks is vulnerable to spoofing attacks as authentication is needed when a user logs into the social network, an attacker could use this type of attack

10.3. Tempering with data

Attacker could change the data in the social network when gain access to the database. Social networks are vulnerable to these attacks

10.4. Information disclosure

Social network are vulnerable to information disclosure threats as users must not gain access into the application without a valid authentication, i.e. user name and password should be the same.

11. Conclusion

Social networks should be protected against attacks as the population of theses environments grow and the potential for attacks aimed at accessing personal information increase in these environments. Organizations are using social networking web applications both as a method of interaction with a wide range of users and also as an environment to sell and advertise their products. Yet this same environment allows users to upload personal information that they want to share with friends and family. Information from this application should be protected against attackers. It is important to model an application because it helps you identify treats and vulnerabilities in the social network. Applying the threat modeling process will help reduce threats in an application. Techniques like the STRIDE model helps in categorizing threats which makes it easier to circumvent attacks. Threat modeling does not guarantee safety of data in a web application but it significantly reduces risks of attacks in an application.

12. References

[1]. http://searchsecurity.techtarget.com/sDefinition/0,290660,sid14_gci1166533,00.html

[2]. http://www.infosecwriters.com/text_resources/pdf/AShrivastava_Web_Application_Threat_Modeling.pdf

[3]. http://www.sans.org/reading_room/whitepapers/securecode/threat-modeling-process-ensure-application-security_1646

[4]. http://www.praetorian.com/presentations/Praetorian_Threat_Modeling_Presentation.pdf

[5]. Jie Wang, Raphael C.-W. Phan, John N. Whitley and David J. Parish High Speed Networks Research Group Department of Electronic and Electrical Engineering, Loughborough University, LE11 3TU, UK Email: {J.Wang3, R.Phan, J.N.Whitley, D.J.Parish}@lboro.ac.uk Augmented Attack Tree Modeling of SQL Injection Attacks

[6]. https://www.owasp.org/index.php/Top_10_2010-Main

[7]. S. Madan and S. Madan. Shielding Against SQL Injection Attacks Using ADMIRE Model. In First International Conference on Computational Intelligence, Communication Systems and Networks, 2009. CICSYN '09., pages 314–320, July 2009.

[8]. JFlow: Practical Mostly-Static Information Flow Control Andrew C. Myers Laboratory for Computer Science Massachusetts Institute of Technology

[9]. http://news.cnet.com/8301-27080_3-20012115-245.html

[10]. http://www.elgg.org/

[11]. http://www.slideshare.net/sensepost/corporate-threat-modelling

[12]. An Approach To Web Application Threat Modeling By Akash Shrivastava April 2008

[13]. http://msdn.microsoft.com/en-us/magazine/cc163519.aspx

[14]. http://www.blackhat.com/presentations/bh-europe-08/Alonso-Parada/Whitepaper/bh-eu-08-alonso-parada-WP.pdf

[15]. http://www.internetworldstats.com/emarketing.htm

[16]. http://msdn.microsoft.com/en-us/library/ff648006.aspx

[17]. http://blogs.msdn.com/b/larryosterman/archive/2007/09/04/threat-modeling-again-stride.aspx

[18]. http://teck.in/stride-model-of-threat-categories.html

[19]. http://msdn.microsoft.com/en-us/library/ff648644.aspx

[20]. Foundations of Attack Trees Sjouke Mauw1 and Martijn Oostdijk21 Eindhoven University of Technology (sjouke@win.tue.nl)