



**Abertay
University**

Threat model comparison

A comparison between STRIDE and PASTA

Jordan Gribben, 1701775

CMP320: Ethical Hacking 3 Mini Project

BSc Ethical Hacking Year 3

2020/21

Abstract

This paper aims to investigate threat modeling methodologies and how they compare against each other. The threat modeling methodologies chosen were Spoofing, Tampering, Repudiation, Information disclosure, Elevation of privilege (STRIDE) and Process for attack simulation and threat analysis (PASTA).

This was done by creating threat models using both methodologies for two web applications and a smart speaker, creating six models in total. Doing this provided a list of threats along with mitigations for each product under both models. These completed models were then compared together by examining which one picked up on more threats.

The PASTA methodology was found to be more affective at discovering more threats due to its research step and use of the attack trees technique, however, it was concluded that more data is needed due to the two web applications being similar in structure. It was also concluded that more data should be gathered to gain a greater comparison.

Contents

1	Introduction	1
1.1	Background	1
1.2	Threat Modeling Techniques	2
1.2.1	STRIDE	2
1.2.2	PASTA	3
1.2.3	Security Cards.....	3
1.2.4	Attack Trees	3
1.2.5	OCTAVE	4
1.3	Aims.....	4
2	Procedure.....	5
2.1	Overview of Procedure	5
2.2	Data Flow Diagrams	5
2.3	STRIDE	7
2.4	PASTA	7
2.5	Shopping Web Application.....	7
2.5.1	Data Flow Diagram.....	7
2.5.2	STRIDE	9
2.5.3	PASTA	12
2.6	Social Media Web Application	17
2.6.1	Dataflow diagram.....	17
2.6.2	STRIDE	19
2.6.3	PASTA	21
2.7	Smart Speaker	26
2.7.1	Data flow diagram.....	26
2.7.2	STRIDE	28
2.7.3	PASTA	29
3	Results.....	33
3.1	Results for Shopping Web Application	33
3.2	Results for Social Media Web Application	33
3.3	Results for Smart Speaker	33
3.4	Overall Results	34

4	Discussion.....	35
4.1	General Discussion.....	35
4.2	Attack trees	35
4.3	Conclusions	35
4.4	Future Work.....	36
5	References	37

Table of Tables

Table 1 - Dataflow Diagram Key.....	6
Table 2 - Spoofing: Shopping Web Application.....	9
Table 3 - Tampering: Shopping Web Application	9
Table 4 - Repudiation: Shopping Web Application	10
Table 5 - Information Disclosure: Shopping Web Application.....	10
Table 6 - Denial of Service: Shopping Web Application.....	11
Table 7 - Elevation of Privilege: Shopping Web Application	11
Table 8 - Spoofing: Social Media Web Application	19
Table 9 - Tampering: Social Media Web Application	19
Table 10 - Repudiation: Social Media Web Application.....	20
Table 11 - Information Disclosure: Social Media Web Application	20
Table 12 - Denial of Service: Social Media Web Application	21
Table 13 - Elevation of Privilege: Social Media Web Application	21
Table 14 - Spoofing: Smart speaker	28
Table 15 - Tampering: Smart speaker	28
Table 16 - Repudiation: Smart speaker.....	28
Table 17 - Information disclosure: Smart speaker.....	28
Table 18 - Denial of service: smart speaker.....	29
Table 19 - Elevation of privilege: smart speaker.....	29

Table of Figures

Figure 1 - Data Flow Diagram: Shopping web application	8
Figure 2 - Attack tree: Gaining admin panel access	15
Figure 3 - Attack tree: gaining access to database information	16
Figure 4 - Dataflow diagram: Social media web application.....	18
Figure 5 - Attack tree: Gaining sensitive information about a user	24
Figure 6 - Dataflow diagram: Smart speaker	27
Figure 7 - Attack tree: Intercepting speaker communications	31

1 INTRODUCTION

1.1 BACKGROUND

When creating any product such as a web application, a smart home device or a type of software it is extremely important to identify any threats that may occur as soon as possible. This is to prevent vulnerabilities and corresponding exploits from being discovered after the release of the product. To catch potential threats early a process known as threat modeling can be used, this process allows for threats to be identified before any development has occurred.

Threat modeling allows for potential threats to be discovered based off the plans of the product. By looking at the technologies used it can be determined what areas may be exposed to potential threats. By discovering threats before development, it allows the areas with the most threats to be focused on to ensure they are as protected as possible.

While threat modeling can be conducted before development it is important to have threat models reviewed consistently. This is to ensure that if any new potential threats appear or if one has been missed from a previous model they can be found, and countermeasures can be put in place before the threat can be exploited.

A completed threat model could be shown to people associated with the product, such as clients and executives, to help them understand why cyber security is an area that should be focused on during development. A threat model should be able to successfully convey the potential threats faced, what this threat means and how it can be mitigated.

While there is conflicting research regarding the method in which threat modelling is conducted, there are various techniques and methodologies available to help identify threats within a product. Due to the various approaches that can be taken when threat modeling, some techniques or methodologies may not be as effective at discovering threats than another.

This report will investigate STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Elevation of privilege) and PASTA (Process for attack simulation and threat analysis) methodologies for creating threat models to discover if one is more effective at discovering potential threats than the other. This will be done by creating threat models for three products using both methodologies, the findings of each model will then be compared to discover if one of the methodologies is more effective than the other.

1.2 THREAT MODELING TECHNIQUES

While there is conflicting research regarding the way in which threat modelling is conducted, there are various methodologies and techniques that can be used to help display a threat model.

1.2.1 STRIDE

Spoofing, Tampering, Repudiation, Information disclosure, Elevation of privilege (STRIDE) is the one of the most used methodologies when threat modeling. STRIDE is a six-step methodology with “STRIDE” being an acronym for each threat type looked at during each stage of the process. The following are the six steps taking within “STRIDE”:

- Spoofing – This stage in STRIDE looks at spoofing threats, this covers threats where a user or program pretends to be someone/thing other than what /who they are, to gain access to otherwise unauthorized areas. For example, a normal user attempts to login as an admin account, IP spoofing, uploading a spoofed file with malicious content, etc.
- Tampering – This stage covers tempering threats, these threats cover modifying data found within areas restricted to that user. This is different from spoofing threats, as a malicious user may not have to spoof their identity to tamper data. However, if a user can gain access to an unauthorized area with a spoofing attack this can lead to tampering. Note, spoofing and tampering may occur simultaneously, however they are not interdependent.
- Repudiation – This stage involves threats where a user claims they are not responsible for an action that has happened. For example, claiming a package never arrived and demanding a refund or a malicious user could claim they had no part in an attack and due to an absence of evidence it cannot be proven otherwise.
- Information Disclosure – This stage involves users being able to access or gain unauthorized access to confidential information. For example, a user may be able to view indexes of various folders and files if they have not been configured properly. A user may also be able to use SQL injection to get information such as usernames and passwords from a database.
- Denial of Service – A denial of service attack involves exhausting a resource so that it can no longer be used for its intended purpose such as, flooding a website with traffic so that it can no longer be accessed by the public. This stage of the methodology looks at any resource within the final product where a denial of service attack could take place.
- Elevation of Privilege – Elevation of privilege involves a user being able to do something they are not authorized to do. This stage of the methodology investigates where a user could exploit this type of attack for example, using SQL injection or cross site scripting to elevate their privilege. This stage in the methodology also ties in with the spoofing and tampering stages as some of the threats covered in those areas may lead to an elevation of privilege.

1.2.2 PASTA

Process for attack simulation and threat analysis (PASTA) is another commonly used threat modelling methodology which follows a seven-step process, the stages within this methodology are as follows:

- Define business objectives – the first step in the PASTA methodology is to define the business objectives, this is done by understanding exactly what the final product will be and what it aims to achieve. These objectives may also be set by external clients.
- Define technical scope – This second step builds upon the first, with the final objective in mind the technologies used must now be defined, such as databases, web servers, and other systems the final product will use.
- Application decomposition – With all the technological components marked out in step two the data flow between these components must be mapped out. The most common way of doing this is with a dataflow diagram.
- Threat analysis – Step four of PASTA involves gathering information of threats that commonly affect the final product as well as the latest threats within that area.
- Vulnerability detection – Building upon steps two and three, step five aims to identify any potential vulnerabilities found within the design of the system. If any code has been developed at this point the code should also be analyzed for potential vulnerabilities.
- Analyze potential attacks – The sixth step taken within PASTA aims to analyze any of the potential threats discovered within the previous sections. Attack trees can be used to analyse these vulnerabilities to discover any potential attacks that come with them.
- Impact analysis – The final step in PASTA uses all the information from previous stages to create a list of countermeasures for all the potential threats discovered.

1.2.3 Security Cards

Security cards are an interesting and engaging way to gamify threat modeling. Security cards often work by splitting types of threats into different suits e.g. Spoofing/Hearts, Tampering/Diamonds, etc. Each card is then given a possible scenario based on their suit, players take turns playing their cards and discussion each scenario as its played. If the scenario brought up is a concern for the active player, they are assigned a point and the threat is noted down. If the scenario played does not cause any concern or does not apply to the product the card is discarded and the player receives no points.

There are many different versions of security cards including “Elevation of Privilege” developed by Microsoft (Microsoft, 2013) and “Cornucopia” developed by OWASP (OWASP, 2021).

1.2.4 Attack Trees

Attack trees are one of the oldest forms of threat modeling, this method uses a tree diagram to convey each potential threat. The ‘root’ of the chart acts as the final goal for an attack and the branches act as potential ways to reach this goal.

While attack trees used to be the most common form of threat modeling it is now rarely used by itself as other methodologies such as STRIDE are more effective. However, while not used on its own it can be used alongside methods such as STRIDE and PASTA to help create a more thorough threat model. By using the potential threats discovered while using STRIDE as the roots and potential ways to get there as the branches a more comprehensive threat model can be established.

1.2.5 OCTAVE

OCTAVE is a threat modeling technique that has three main phases. The first phase involves building a threat profile. This profile evaluates what assets within the product or organization are deemed important and establishing which of these assets are most critical. With the most critical assets established the threats for each asset can then be looked at from the most critical to the least. The second phase involves analyzing potential threats with the products infrastructure such as the dataflows and creating a plan on how to mitigate these issues. The final phase of OCTAVE involves gathering the threats and critical assets discovered from the first two phases and prioritizing the mitigations based on how critical the threats could be. Once the threat order has been noted down a final report on the changes that have to be made can be created and implemented.

1.3 AIMS

This project aims to complete the following objectives:

- Create threat models for three separate products
- Follow the STRIDE and PASTA methodologies for the threat models
- Compare the threat models created by each methodology
- Discover if one threat modelling methodology is more effective than the other

2 PROCEDURE

2.1 OVERVIEW OF PROCEDURE

Three products will be used to create the threat models, these products are:

- Shopping web application
- Social media web application
- Smart speaker

Each product will have a threat model created using both the STRIDE and PASTA methodologies alongside a brief description of the product, this will be done by first creating a dataflow diagram for each product to visualize how the products work and how they carry data. These diagrams will also help identify where within the products; a potential threat could occur.

Once a dataflow diagram has been created the STRIDE methodology will first be used to create a threat model of the product. When using the STRIDE methodology, the threats will be placed into tables based on the threat type. The tables will include, the potential threat, name of attack used for that threat, a description of that attack and where within the dataflow diagram the threat could occur. Once all the potential threats have been listed, mitigations for these threats will be discussed so that they can be implemented during development to avoid the threats completely.

With the STRIDE threat model completed, a new threat model will be created using the seven-step pasta methodology. The first stage of PASTA covers the business objectives this will align with the description of the product. Stages two and three investigate the technologies used and these components communicate with each other, while some technologies such as web servers will be listed separately these two stages will be covered by the dataflow diagram. The sixth step requires an analysis of the attacks due to there not being a physical product to test this on this will be done using the attack trees technique. These attack trees will look at the most serious potential attacks for each product.

Each threat model requires mitigations to be stated however, to avoid repetition only new threats discovered within each model will be mitigated.


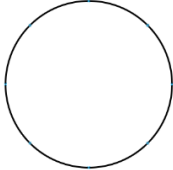
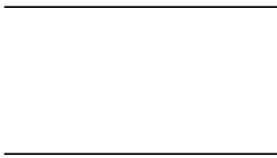


Once both threat models have been developed for each product, the models can be compared to discover if one model is more effective over the other.

2.2 DATA FLOW DIAGRAMS

Before a threat model can be created a dataflow diagram (DFD) must first be created, these diagrams are created so there is a visual representation of how the data the product uses flows and is processed. By visually representing the product, where threats can occur can be more easily identified as where the data is going can be physically seen. For example, if there is an item search that sends data to a database a threat may occur here as a malicious user may try to inject code into the database.

Dataflow diagrams contain various elements representing different functionality's, the following table in table 1 is the key used by all the data flow diagrams.

Table 1 - Dataflow Diagram Key

Element	Name of Element	Description
	External Entity	External entities are often where data is inputted and outputted. Due to this external entities tend to be client machines, people, invoices etc.
	Process	A process is an activity that can change/transform the flow of data, due to this a process must have an input and an output. For example a client machine as an external entity could input two numbers to a process labeled "addition" this process would then have a dataflow back to the client machine external entity with the data now processed to be the result of the two numbers being added together.
	Datastore	Datastores hold data for later access, one common way this is done is via databases. These could be used to store things such as user/product data.
	Data flow	Data flow shows the direction the data is flowing in. It can be used to show what information is going into which element.
	Trust boundaries	Trust boundaries help note areas of concern, by indicating exactly where trust level changes. For example, if data is being inputted to somewhere such as a login, can this input be trusted.

To create the dataflow diagram, the online tool diagrams.net (Diagrams.net, 2021) will be used.

2.3 STRIDE

The STRIDE methodology will be used alongside the dataflow diagrams to create a threat module for each product. This will be done by creating tables for each stage of the methodology, the tables will contain: Where on the dataflow diagram the threat could occur, what the potential threat is, names of potential attacks for that threat, and a description of that threat. With all the potential threats listed mitigations will then be discussed.

Each threat was discovered by viewing the dataflow diagram and questioning what elements could be threatened and how this threat could be carried out. With the trust boundaries helping to identify where these could occur due to the data being passed to each area.

Attack trees will not be used alongside the STRIDE methodology as it is not a key component within this methodology.

2.4 PASTA

After a threat model has been created using the STRIDE methodology a second threat model will be created using the PASTA methodology. This will be done by following each step as close as possible, step one will outline the business objectives defining exactly what each product sets out to do. Step two will define the technologies used such as how the web application will be hosted, what type of database is used as well as the process used to make the product work. Step three will be the dataflow diagram created at the start of the process this is due to step three defining how the technologies and process communicate with each other. During the fourth stage of PASTA the common threats faced by the products will be researched, these threats will first be listed and then described. The threats found within step five will also be investigated during this stage to see if they can be applied to the product. Step six involves the threats to be analyzed, this will be done by taking the two most critical threats and creating attack trees to discover how these threats could be approached. With the main threat model now complete step seven will review all the potential threats and attacks found and give mitigations for each so that during development they can be implemented.

2.5 SHOPPING WEB APPLICATION

A Threat model will be created for a basic shopping web application. This application will allow users to create their own accounts, search for items, purchase items, write reviews and apply for returns.

2.5.1 Data Flow Diagram

In figure 1 the dataflow diagram for the shopping web application can be seen. It contains one external entity, three datastores and various processes. This diagram shows that the shopping web application will contain the features listed in section 2.5

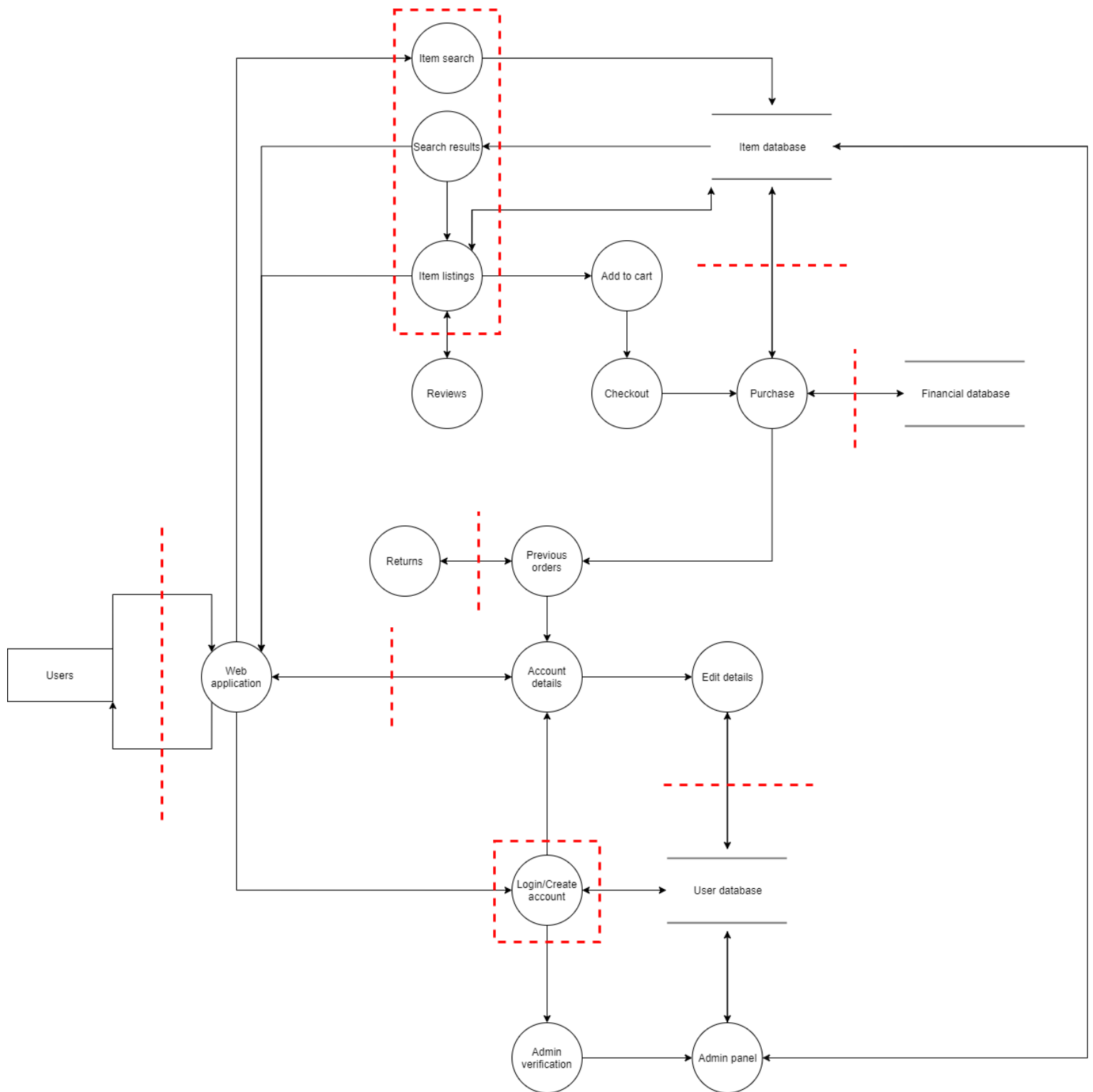


Figure 1 - Data Flow Diagram: Shopping web application

2.5.2 STRIDE

The dataflow diagram was examined for potential threats these threats were then noted, and the following threat model was created.

Table 2 - Spoofing: Shopping Web Application

Element	Threat	Spoofing attack	Threat description
Admin verification	Spoofing as an admin account	SQL injection, Brute force	An attacker could potentially spoof the admin verification to gain access to the admin panel by using SQL injection to log into an administrative account.
Web application	Reading traffic	Man in the middle attack	A malicious user could potentially use ARP spoofing to view incoming network traffic.
Reviews, Returns, Web application	Session hijacking	Session hijacking, Cross site scripting	A malicious user could potentially take advantage of cross site scripting in order to gain access to another users session, this could be potentially dangerous as if done in the returns section the user may be able to gain access to an admin session.
User	IP spoofing	IP spoofing	If an IP has been banned from the web application a user could potentially spoof their IP to get around this ban.

Table 3 - Tampering: Shopping Web Application

Element	Threat	Tampering attack	Attack description
Item database	Modifying items	SQL injection, Cross site scripting	A user could potentially use SQL injection or cross site scripting to modify items within the databases, this could happen if a malicious user manages to gain access to an admin account as seen in section 2.4.2.1. This could also happen without access to an admin account due to the trust boundary around the item search as a SQL injection attack could take place there.
User database	Modifying user data	SQL injection, Cross site scripting	This attack would work the same way as the item database but with the User database instead.

Table 4 - Repudiation: Shopping Web Application

Element	Threat	Repudiation attack	Attack description
Returns	Claims to not have purchased	N/A	A malicious user could potentially claim that they did not purchase an item and that the money just came out their account.
Returns	Claims item did not arrive	N/A	A malicious user could potentially claim that a product they purchased never arrive to receive money back.

Table 5 - Information Disclosure: Shopping Web Application

Element	Threat	Information disclosure attack	Attack description
All datastores	Extract data from databases	SQL injection	By using a SQL injection attack a malicious user could potentially read sensitive information found within the database.
Web application	Extract data from source code	Source code vulnerability	If not examined before publication the source code may contain sensitive information, such as version numbers. This information could aid malicious users in attacking the web application.
Login/create account, item search	Extract data from error messages	User enumeration	If the login system contains specific errors such as 'Incorrect password' an attacker would be able to take advantage of this information as they would know while the password is incorrect the username is correct.
Web application	Get data from potential file listings	Directory browsing	If not configured properly on the webserver users may be able to see file directory's containing sensitive information.

Table 6 - Denial of Service: Shopping Web Application

Element	Threat	Denial of service attack	Attack description
Web application	Exhausting the website	DoS	An attacker could potentially flood the web application with traffic so it can no longer be accessed by general users.
All datastores	Exhausting the databases	DoS	Complex queries could be given to the database causing it to exhaust its resources.

Table 7 - Elevation of Privilege: Shopping Web Application

Element	Threat	Elevation of privilege attack	Attack description
Admin verification, edit details	Spoofing an admin account	SQL injection, Cross site scripting	If an attacker successfully manages to spoof into a user account, they will have successfully elevated their privilege reaching an area they should not be authorized to enter. Additionally, as seen in section 2.4.2.2 a malicious user may be able to tamper with the admin panel to elevate their accounts privilege.

2.5.2.1 Mitigations

With the potential threats listed and placed within respective tables mitigations can now be looked at. This will ensure that during development the mitigations can be implemented making the final product more secure. The following are the mitigations based off the created threat model:

- Spoofing – To avoid brute force attacks a strong password policy should be enforced for the administrators and heavily advised for the customers. Limiting the number of login attempts before a user is timed out can also help prevent brute force attacks. HTTPS should be enforced on the web application, to ensure the communication between the application and user is secure. In doing this it will help prevent a man in the middle attack. Packet filtering could be used to help prevent IP spoofing, this method will check for any inconsistencies such as source IP's not matching and block this traffic.
- Tampering – To avoid SQL injection any input that leads to a database should be striped of any characters related to SQL injection before being sent to the database. The X-XSS-Protection header should be defined within the web application to help protect against cross site scripting attacks.
- Repudiation – Logs must be kept of items purchased by each user along with delivery logs. This way, if a malicious user makes a false claim, the company will have the correct logs to back prove themselves.

- Information disclosure – The source code must be analyzed before publication to ensure no sensitive information can be found; the code could also be reviewed by multiple people to decrease the chance of any sensitive information being left behind. Any error message presented to a user should be generic to ensure no information can be gained from these messages. Once set up the webserver must be told not to display the directories to mitigate the potential directory browsing threat.
- Denial of service – One way to mitigate against a denial of service attack would be to block specific IP's from the site. While extreme if the traffic from the DoS attack can be traced to a specific area all the IP's within that area can be blocked, this allows legitimate users from other areas access to the site.
- Elevation of privilege – Any potential attacks to elevate a user's privilege has been covered in the previous mitigations such as SQL injection and cross site scripting.

2.5.3 PASTA

2.5.3.1 *Define Business Objectives*

The business objective is to create a basic shopping web application that allows users to create an account, search for items, purchase items, leave reviews and file for returns.

2.5.3.2 *Define technical scope*

With the business objective outlined the technical scope can now be created, the following are the essential technical components.

- WordPress web server to host web application
- MySQL database for users, items, and financial database.
- Various processes allowing for key functionality such as an item listing, item search's, reviews, returns and user login.

2.5.3.3 *Application decomposition*

The dataflow diagram in figure 1 shows the components within the technical scope communicate with each other.

2.5.3.4 *Threat analysis*

Despite online shopping being part of everyday life, it still faces various potential threats. The following are common threats found within the online retail world:

- DoS attacks – These attacks can be used so that the website will become unavailable to users and no sales can be made.
- Bank card fraud – If a malicious user uses a stolen bank card's details to make a purchase within the shopping web application a serious act of fraud has been committed. The bank details could have been stolen from data breaches, a malicious user accessing another person's account additionally, the card could have also been stolen in person.

- SQL injection – If a SQL injection attack is successful an attacker could be able to; view information for databases, access other users accounts and login as an admin.
- Cross site scripting – An attacker could use cross site scripting to place malicious code within the web application. For example, an attacker could leave a review that contains malicious code so that if a user goes onto that page, they could view that user's sensitive information.
- Point of sale attacks – Attacks that target the point of sale systems are extremely dangerous as any weakness in this system could allow for a user's banking information to be exposed.
- Bots – Bots can be used to scrape websites for information, for the shopping web application they may be used to get pricing information for competitors, or a malicious user may use it to find vulnerabilities.
- Phishing – Customers can often face phishing attacks; this is often done by an attacker sending a convincing looking email to a customer that sends them to a near identical site to steal their details. This could potentially put customers off using the site especially if their email address was discovered due to a data breach from the shopping web app.
- Brute force – Many attackers will attempt brute force attacks to try log in as an admin account so they can gain access to confidential information. This is especially dangerous if the administrators have weak passwords.
- Man in the middle – An attacker may use a man in the middle attack in order to sniff the network traffic. Doing this could reveal sensitive information such as passwords, bank details and session ID's.
- Malware – Rather than attack the site directly a user or administrator may have malware on their PC. Depending on the type of malware an attacker may be able to use this to gain access to user accounts, this is especially dangerous if this happens to an administrator.

2.5.3.5 Vulnerability detection

With the common threats now known, the applications design and code can now be examined for vulnerabilities. Due to there being no code to review only the applications design will be looked at however, it is important to examine code for any sensitive information before publication.

In section 2.4.3.2 it is stated that a wordpress sever will be used to host this web application, it is important to read the version notes and use the most up to date version of wordpress to ensure the most protection from threats. This is due to threats for older versions being commonly known and easily exploited.

Taking the common threats from the threat analysis section, it is determined that all the threats listed are possible to occur with the current application design. The weakest points in the applications design is currently the login / admin verification and the datastores, this is due to the amount of information and

control an attacker would gain if they successfully attacked these sections. The common attacks listed also show this as six of them revolve around gaining user credentials.

2.5.3.6 *Analyze potential attacks*

Using attack trees two main attacks will be analyzed, only two will be analyzed due to overlap within attacks. For these attack trees the 'root' will be the end goal of the attack with the branches being potential ways to get there. The attack trees for gaining admin panel access alongside the attack tree for gaining access to database information can be seen in figures 2 & 3.

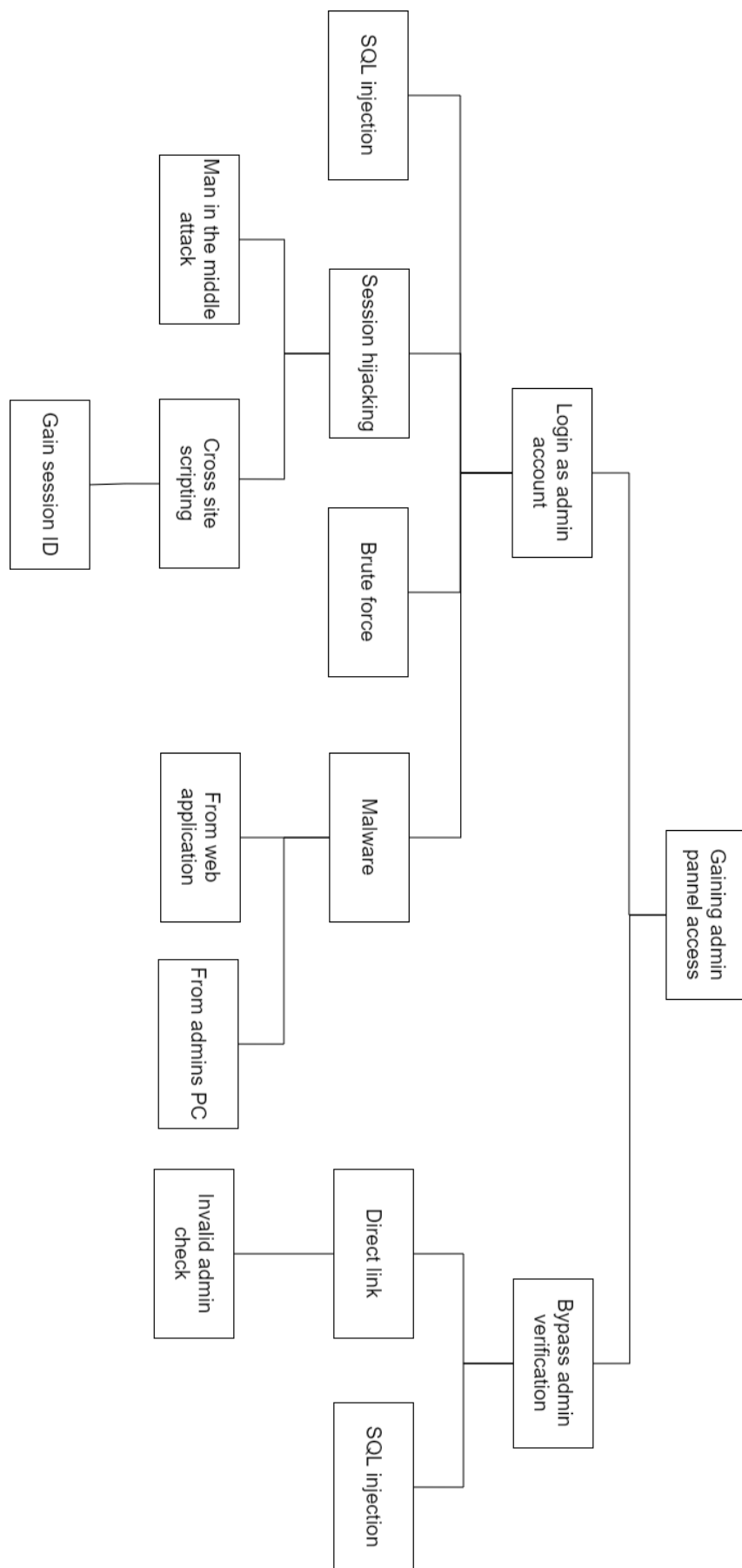


Figure 2 - Attack tree: Gaining admin panel access

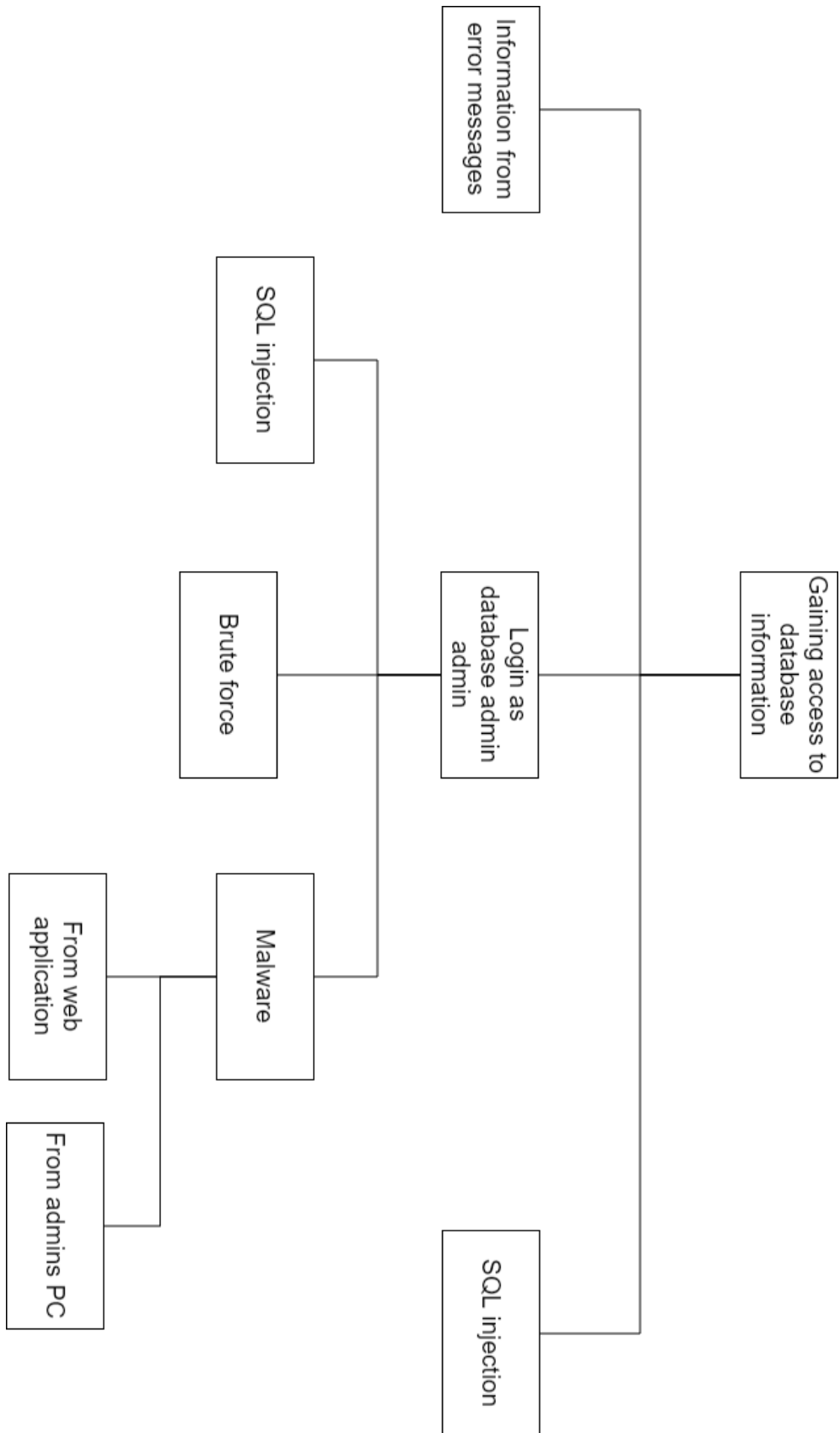


Figure 3 - Attack tree: gaining access to database information

2.5.3.7 *Impact analysis*

As a list of mitigations can be found in section 2.5.2.1 for the STRIDE methodology, only mitigations for new threats will be covered in this section. The mitigations within section 2.5.2.1 can still be applied to the some of the threats found under the PASTA methodology. The following are mitigations for the new threats found under PASTA:

- **Malware** – A strong antivirus system should be in place on the website to help protect against any malware trying to be uploaded to the site. If malware was to make it on to the website the antivirus should be able to pick up on this, remove the malware and alert the administrators. Administrators should also regularly check their own devices for any form of malware to ensure an attack won't be able to access their system.
- **Valid authentication** – A valid authentication process must be in place to ensure only authorized users can access specific areas, even if a direct link is used the restricted areas should all feature an authentication check.
- **Phishing** – Users should be made aware of the exact situations they could be emailed from the shopping application. Users should also access the website directly rather than clicking on any links sent by email. In doing this it should help prevent against phishing scams.
- **Bank card fraud** – An address verification service should be used on the web application to help prevent against fraud.

2.6 SOCIAL MEDIA WEB APPLICATION

A Threat model will be created for a basic social media web application. This application will allow users to create an account, edit their profile picture make posts, like / comment / share posts, and search for posts.

2.6.1 Dataflow diagram

In figure 4 the dataflow diagram for the social media web application can be seen. It contains one external entity, two datastores and various processes. This diagram shows that the social media web application will contain the features listed in section 2.6

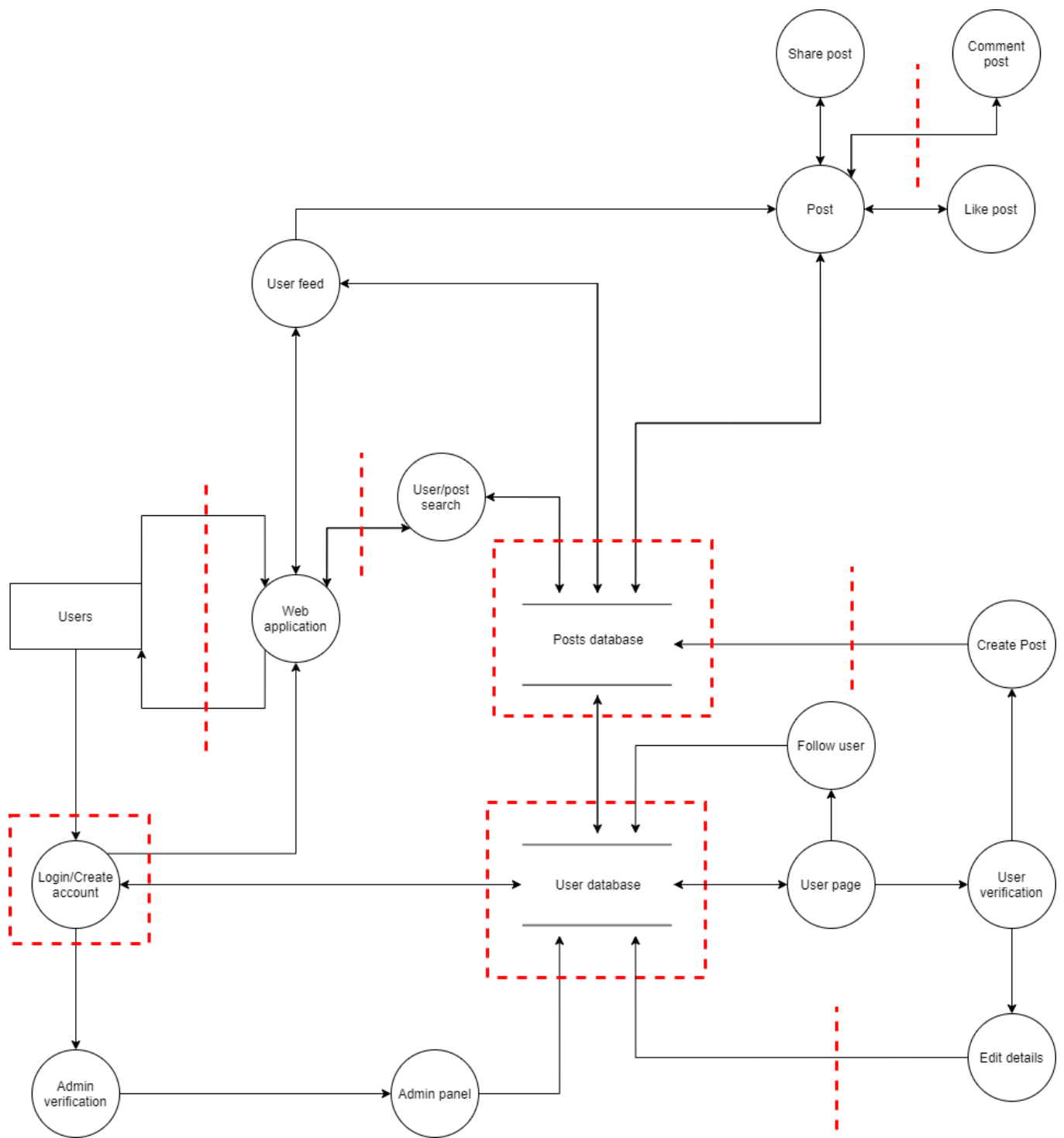


Figure 4 - Dataflow diagram: Social media web application

2.6.2 STRIDE

The dataflow diagram was examined for potential threats, these threats were then noted, and the following threat model was created.

Table 8 - Spoofing: Social Media Web Application

Element	Threat	Spoofing attack	Threat description
Admin verification	Spoofing as an admin account	SQL injection, brute force	An attacker could potentially spoof the admin verification to gain access to the admin panel by using SQL injection to log into an administrative account.
Web application	Reading traffic	Man in the middle attack	A malicious user could potentially use ARP spoofing to view incoming network traffic.
Web application, post	Session hijacking	Session hijacking, Cross site scripting	A malicious user could potentially take advantage of cross site scripting in order to gain access to another users session, this could be potentially dangerous as if done in the returns section the user may be able to gain access to an admin session.
User	IP spoofing	IP spoofing	If an IP has been banned from the web application a user could potentially spoof their IP to get around this ban.
Post, User page	File spoofing	File upload vulnerability	When uploading a picture to post or changing profile picture an attacker could potentially upload a spoofed file that could contain malware.

Table 9 - Tampering: Social Media Web Application

Element	Threat	Tampering attack	Attack description
Post database	Modifying posts	SQL injection, Cross site scripting	A user could potentially use SQL injection or cross site scripting to modify items within the databases, this could happen if a malicious user manages to gain access to an admin account as seen in section 2.5.2.1. This could also happen without access to an admin account due to the trust boundary around the item search as a SQL injection attack could take place there.
User database	Modifying user data	SQL injection, Cross site scripting	This attack would work the same way as the item database but with the User database instead.
Web application	File tampering	File upload vulnerability	As mention in section 2.5.2.1 the social media web application could end up having a file upload vulnerability. If a file with malware is uploaded it has the potential to overwrite, corrupt or delete critical files the website uses.

Table 10 - Repudiation: Social Media Web Application

Element	Threat	Repudiation attack	Attack description
Web application	Editing logs	Insufficient controls	A malicious user could potentially change the web applications logs, if this is done, they could potentially deny having any form of involvement in an attack or make it look like another user is responsible.

Table 11 - Information Disclosure: Social Media Web Application

Element	Threat	Information disclosure attack	Attack description
All datastores	Extract data from databases	SQL injection	By using a SQL injection attack a malicious user could potentially read sensitive information found within the database.
Web application	Extract data from source code	Source code vulnerability	If not examined before publication the source code may contain sensitive information, such as version numbers. This information could aid malicious users in attacking the web application.
Login/create account, item search	Extract data from error messages	User enumeration,	If the login system contains specific errors such as 'Incorrect password' an attacker would be able to take advantage of this information as they would know while the password is incorrect the username is correct.
Web application	Get data from potential file listings	Directory browsing	If not configured properly on the webserver users may be able to see file directory's containing sensitive information.

Table 12 - Denial of Service: Social Media Web Application

Element	Threat	Denial of service attack	Attack description
Web application	Exhausting the website	DoS	An attacker could potentially flood the web application with traffic so it can no longer be accessed by general users.
All datastores	Exhausting the databases	DoS	Complex queries could be given to the database causing it to exhaust its resources.

Table 13 - Elevation of Privilege: Social Media Web Application

Element	Threat	Elevation of privilege attack	Attack description
Admin verification, edit details	Spoofing an admin account	SQL injection, Cross site scripting	If an attacker successfully manages to spoof into a user account, they will have successfully elevated their privilege reaching an area they should not be authorized to enter. Additionally, as seen in section 2.4.2.2 a malicious user may be able to tamper with the admin panel to elevate their accounts privilege.

2.6.2.1 Mitigations

Due to the shopping web application and the social media web application sharing near identical threats under STRIDE, the same mitigations apply here. See section 2.5.2.1 for mitigations.

2.6.3 PASTA

2.6.3.1 Define Business Objectives

The business objective is to create a basic shopping web application that allows users to create an account, edit their profile picture make posts, like / comment / share posts, and search for posts.

2.6.3.2 Define technical scope

With the business objective outlined the technical scope can now be created, the following are the essential technical components.

- WordPress web server to host web application.
- MySQL database for users and posts.
- The web application will have various processes that allow users to do the features lined out within the business objective.

2.6.3.3 Application decomposition

The dataflow diagram in figure 4 shows the components within the technical scope communicate with each other.

2.6.3.4 *Threat analysis*

Social media applications are used every day, with millions of users logging on each day. With this it is vital that social media sites keep its users protected from any potential threats. The following are common threats found within social media platforms:

- **Phishing** – Social media users can often encounter phishing attacks, by sending a convincing looking email to a client warning them that their account has been suspended the email would then send them to a near identical site where they details could be stolen. This could potentially put customers off using the site especially if their email address was discovered due to a data breach from the social media application.
- **Fake accounts/catfishing /social engineering**– Social media can often be full of fake accounts; this becomes an issue due to the scams that tend to follow along them. A malicious user could pretend to be another person such as a celebrity within the social media platform, they may then message fans to gain sensitive information.
- **Malware** – Malware could potentially be uploaded onto the web application this could be done in various ways. One way this could be achieved is if a user uploads a new profile picture that contains malicious code, this code could then affect any user who visits the malicious profiles page.
- **Links** – New articles and other links are often shared around on social media; however, it is important to be cautious when clicking on links as it may take a user to malicious content.
- **Data breach** – Due to the large number of users on social media platforms it is important to keep the user database as secure as possible, as it could potentially lead to millions of users having their data stolen.
- **Oversharing** – Users may overshare on social media leading to sensitive information getting revealed by accident, this can include things such as a user's location or even the answers to security questions like "what was your first pets name".
- **Third party applications** – Lots of third party apps for social media apps, these apps link to a user's profile and then give them information such as who has unfollowed them or who the people they talk to most are. While sometimes harmless, linking these apps to an account could potentially be a way to install malicious content onto a user's device or even as a tool to lock them out their own account.

2.6.3.5 *Vulnerability detection*

With the common threats now known, the applications design and code can now be examined to further search for vulnerabilities. Due to there being no code to review only the applications design will be looked at however, it is important to examine code for any sensitive information before publication.

In section 2.6.3.2 it is stated that a wordpress sever will be used to host this web application, it is important to read the version notes and use the most up to date version of wordpress to ensure the most protection from threats. This is due to threats for older versions being commonly known and easily exploited.

Taking the common threats from the threat analysis section, it is determined that all the threats listed are possible to occur with the current application design. The common threats listed in the previous step mostly revolve around tricking users so that sensitive information can be gained or to infect a device with malware.

The weakest points in the applications design is currently the login/admin verification and the datastores, this is due to the amount of information and control an attacker would gain if they successfully attacked these sections.

2.6.3.6 *Analyze potential attacks*

Using attack trees technique getting sensitive user information will be analyzed this can be seen in figure 5. The previous attack trees found in section 2.5.3.6 can also be applied to this threat model.

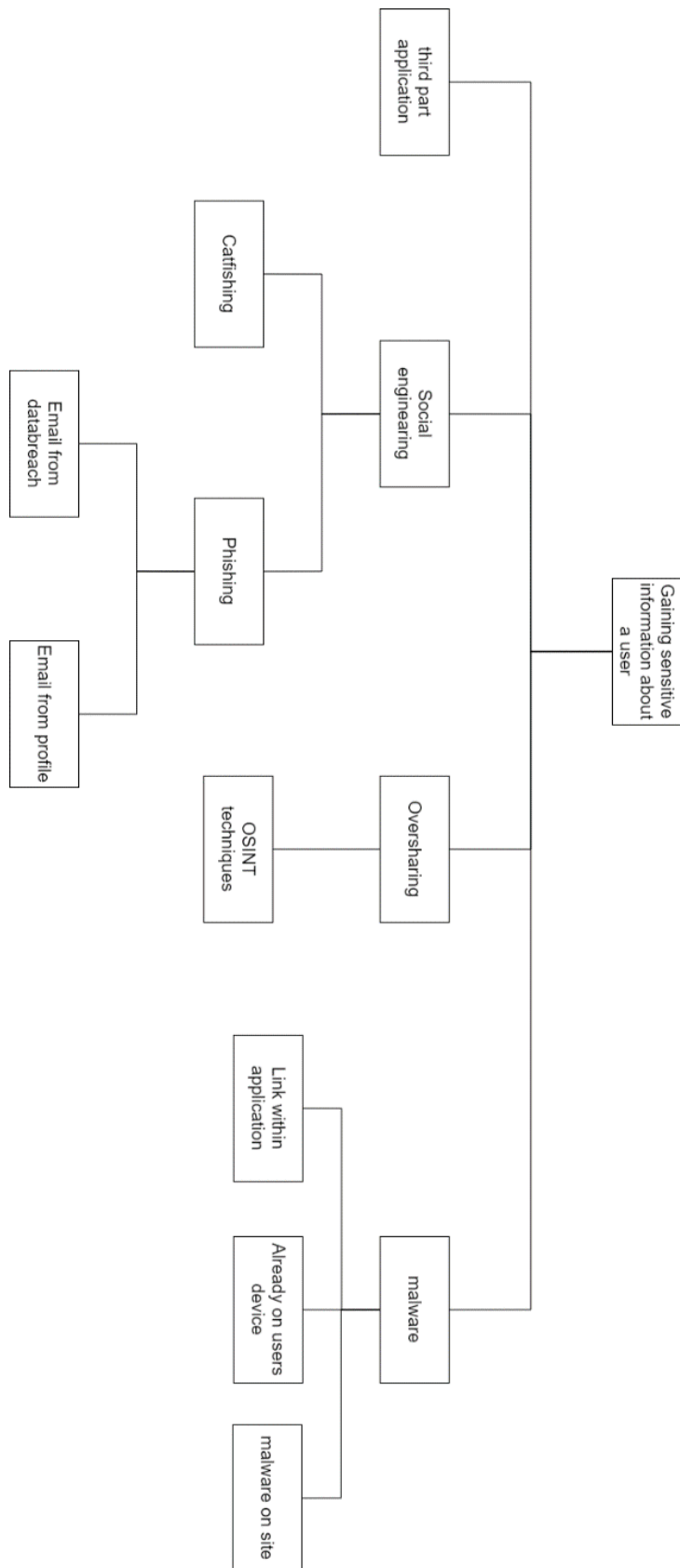


Figure 5 - Attack tree: Gaining sensitive information about a user

2.6.3.7 *Impact analysis*

The following is mitigations for the new threats found for the social media web application using the PASTA methodology:

Social engineering – A verification system should be put in place on the web application to help prevent users from being tricked by fake accounts. Additionally, a report feature should be added so that users can report accounts they think are fake or malicious, this would then allow for the account to be reviewed and then removed from the site if necessary.

Third party apps – Users should be informed that they should only link accounts to third party applications that can be trusted. While this won't prevent malicious third-party apps, it should make users think carefully before signing up to these apps.

Oversharing – Users should be made aware to not share any sensitive information on the social media platform. Additionally, various privacy features such as being able to lock and block accounts should be added so users can choose exactly who sees their content.

Links – User education is a must, by telling users to only click on links they think can be trusted should help reduce the number of users affected. Additionally, a post report button should be added this will allow users to report posts they think contain malicious content. This will allow the content to be reviewed and then removed if necessary.

2.7 SMART SPEAKER

A threat model will be created for an internet of things (IoT) smart speaker. This smart speaker will allow users to create an account on setup, use voice commands to complete actions, receive updates from online, press physical buttons to complete actions such as mute.

2.7.1 Data flow diagram

In figure 6 the dataflow diagram for the smart speaker can be seen. It contains three external entities being the user, the speakers output and the company, three datastores and various processes. This diagram shows that the shopping web application will contain the features listed in section 2.7

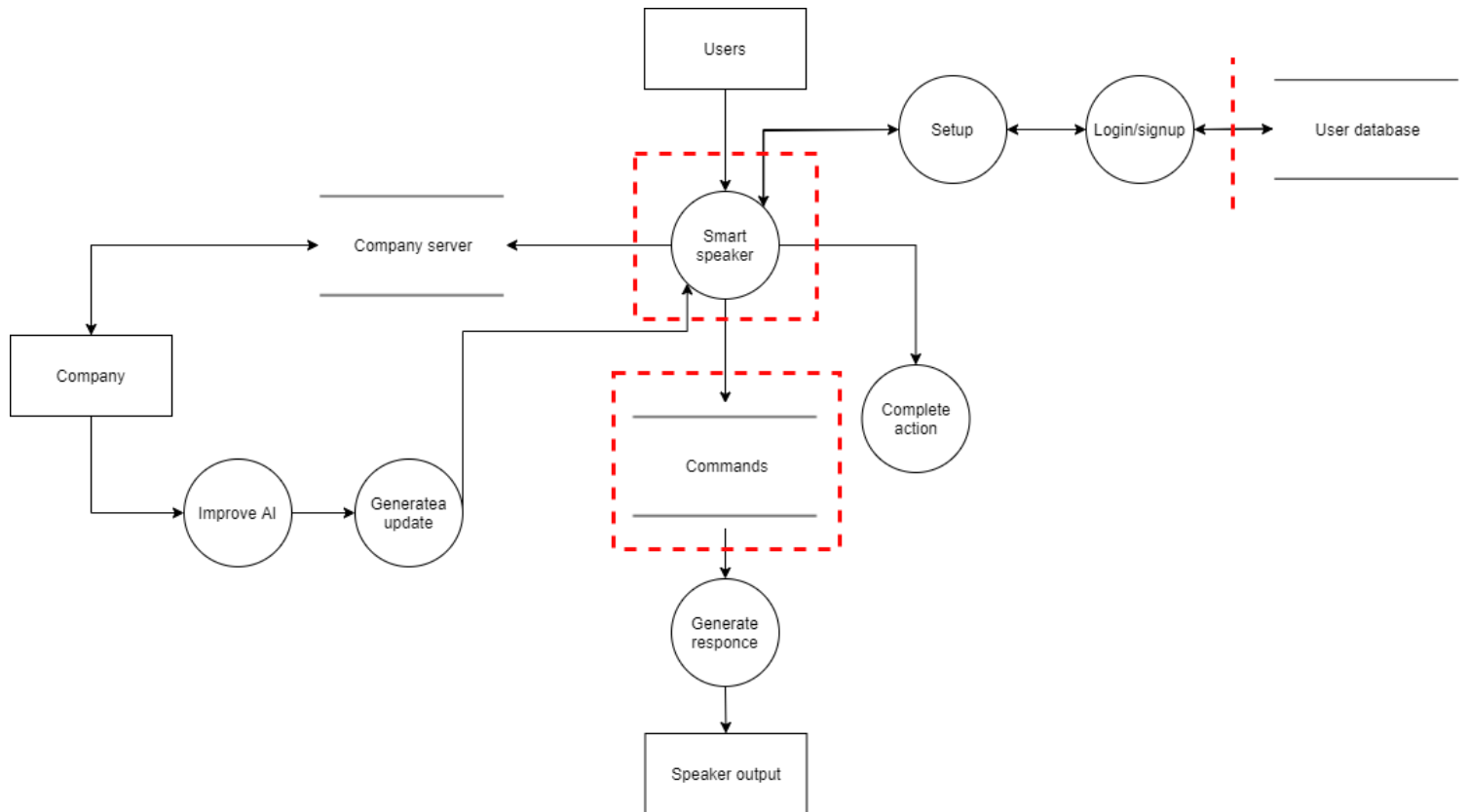


Figure 6 - Dataflow diagram: Smart speaker

2.7.2 STRIDE

The dataflow diagram was examined for potential threats these threats were then noted, and the following threat model was created.

Table 14 - Spoofing: Smart speaker

Element	Threat	Spoofing attack	Threat description
Login/signup	Account spoofing	SQL injection, Brute force	An attacker could potentially spoof into another account by using SQL injection or brute force.

Table 15 - Tampering: Smart speaker

Element	Threat	Tampering attack	Attack description
Smart speaker	Device tampering	Physical tampering, Malware	The physical smart speaker device could be tampered with. By tampering with the physical device, a malicious user may be able to upload malware onto the device and change its intended purpose.
Company server	Data tampering	Physical tampering, Malware	If the source code where to be edited within the smart speaker a malicious user may be able to send malicious code to the company server.

Table 16 - Repudiation: Smart speaker

Element	Threat	Repudiation attack	Attack description
Smart speaker	Device tampering	Physical tampering	A malicious user may try to deny physically altering a device.

Table 17 - Information disclosure: Smart speaker

Element	Threat	Information disclosure attack	Attack description
Smart speaker	Extract data from source code	Source code vulnerability	A malicious user may be able to tamper with the physical device to be able to view the devices source code. If not checked before publication the source code may contain sensitive information.
All datastores	Extract data from databases	SQL injection	By using a SQL injection attack a malicious user could potentially read sensitive information found within the database.

Table 18 - Denial of service: smart speaker

Element	Threat	Denial of service attack	Attack description
Company server, Commands	Exhausting the databases	DoS	A DoS attack could be performed on the company server or the commands list databases, by doing this the smart speaker would no longer be able to perform commands and the company server would not be able to receive legitimate information from the speakers.

Table 19 - Elevation of privilege: smart speaker

Element	Threat	Elevation of privilege attack	Attack description
Smart speaker	Potential access to dev tools	Tampering	If the smart speaker has been tampered with by a malicious user, they may be able to gain access to otherwise inaccessible development tools.

2.7.2.1 Mitigations

The following are mitigations for all new threats found for the smart speaker under the STRIDE methodology:

Physical tampering – While physical tampering cannot be prevented various steps can be taken to discourage users from doing so. One method would be to warn users on the packaging that if this product is tampered with it breaks the products terms of use, a seal could also be placed inside the speaker so that if the speaker were to be opened the seal would break. Devices should also be stored in a secure location to prevent tampering from third parties.

2.7.3 PASTA

2.7.3.1 Define Business Objectives

The business objective is to create a basic internet of things (IoT) smart speaker, this smart speaker will, allow users create an account on setup, use voice commands to complete actions, receive updates from online, press physical buttons to complete actions such as mute.

2.7.3.2 Define technical scope

With the business objective outlined the technical scope can now be created, the following are the essential technical components.

- MySQL database for command list, users, and company server.
- Physical components for the smart speaker device
- The smart speaker will have various processes that allow users to do the features lined out within the business objective.

2.7.3.3 *Application decomposition*

The dataflow diagram in figure 6 shows the components within the technical scope communicate with each other.

2.7.3.4 *Threat analysis*

It is common for a household to have various IoT devices one of the most common is the smart speaker. It is extremely important that these devices are kept secure as they can provide a personal look into a user's home life. The following are common threats faced by these devices:

- Default settings – Lots of devices ship with default settings on, without changing these settings the device might not be as secure as possible. For example, if a smart speaker ships with default settings it may also have a default password, due to this being the default this password would be extremely easy to discover online allowing anyone to access the device.
- Lack of secure update mechanics – It is important devices are kept up to date to keep as safe as possible from security threats. Some devices lack a secure form to update devices, this could be due to the lack of firmware validation. Devices should also notify users of all security changes made to a device after updating.
- Lack of physical hardening – Devices may have a lack of physical hardening, this could potentially allow a user to gain access to sensitive information within the device.
- Insecure network services – IoT devices may leave ports open within a network. Leaving these ports open may lead to the network gaining vulnerabilities, allowing for a malicious user to exploit these vulnerabilities with a variety of attacks.
- Insecure data transfer – Due to IoT devices transferring data from the device to online it is essential that the data transfer is kept secure. If this is not kept secure it a malicious user could easily intercept the traffic.

2.7.3.5 *Vulnerability detection*

With the common threats now known the products design and code can now be examined to further search for vulnerabilities. Due to there being no code to review only the applications design will be looked at however, it is important to examine code for any sensitive information before publication.

Taking the common threats from the threat analysis section, it is determined that all the threats listed are possible to occur with the current products design. The common threats listed in the previous step mostly revolve around how the speaker's security focusing on areas such as secure communication.

The weakest points in the applications design is currently the physical speaker and its communications, amount of information an attacker would gain if they successfully attacked these sections.

2.7.3.6 *Analyze potential attacks*

Using attack trees two main attacks will be analyzed. The attack tree for intercepting speaker communication can be seen in figure 7.

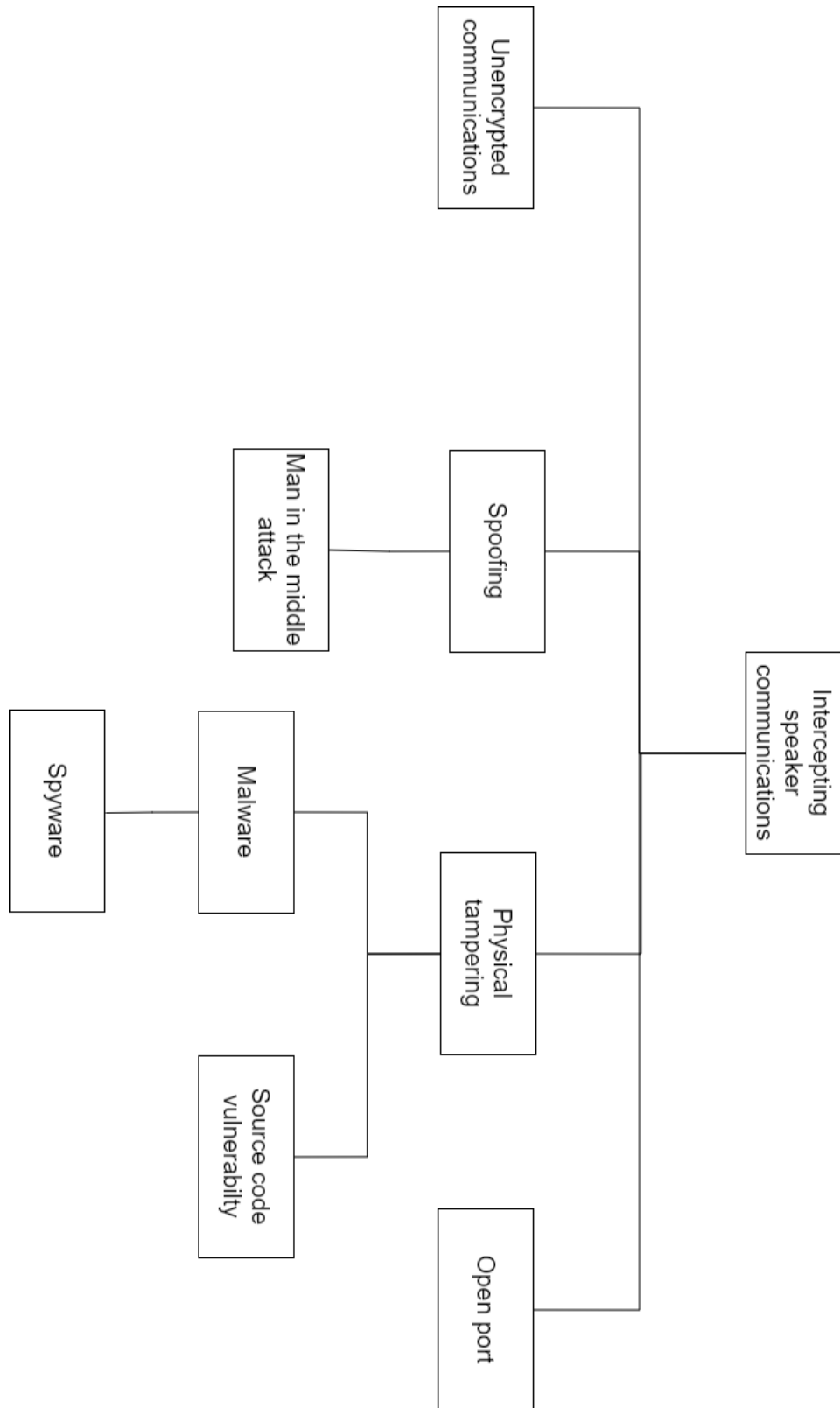


Figure 7 - Attack tree: Intercepting speaker communications

2.7.3.7 *Impact analysis*

- Open ports – Only essential ports should be open, all other open ports should be closed to help prevent against vulnerabilities.
- Default settings – All default settings should be changed on setup to ensure devices are kept secure. By changing default settings such as the default password the it will increase the devices overall security.

3 RESULTS

3.1 RESULTS FOR SHOPPING WEB APPLICATION

Both threat models managed to identify different types of threats that could affect the shopping web application, while different threats were discovered between the methodologies the types of attacks used for these threats are extremely similar.

Due to the lack of source code to review when using the PASTA threat model attacks such as directory browsing could not be covered using this threat model compared to STRIDE. However, while STRIDE could pick up on these threats it was not able to pick up on threats such as bank card fraud, bots, or threats with specific technologies such as word press servers.

Overall PASTA managed to pick up on more threats compared to STRIDE for this application.

3.2 RESULTS FOR SOCIAL MEDIA WEB APPLICATION

Once again both threat models managed to identify different types of threats that can affect the social media web application.

While the STRIDE model remained largely similar when compared to the one found for the shopping web application, the PASTA model differs during both the research and analysis stage. This is in line with the previous findings as PASTA differed during the research stage during the shopping web application.

PASTA managed to pick up on threats involving a more human element due to the research phase alongside the attack trees. The attack trees greatly help when using the PASTA method as it allows for multiple approaches to be viewed based off one threat outcome. These trees allow for the attacks mentioned within STRIDE to be picked up by PASTA if they weren't during a previous stage.

Once again PASTA has managed to pick up on more threats when compared to STRIDE.

3.3 RESULTS FOR SMART SPEAKER

By comparing both threat models made for the smart speaker it can be seen that PASTA has discovered more potential threats when compared to STRIDE. However, PASTA has missed out on some essential threats it has previously picked up on such as SQL injection and DoS attacks.

Both methodologies can discover the possibility of physically tampering with the device. Previously STRIDE has been unable to pick up on more 'human element' threats such as fraud and oversharing.

Overall PASTA has discovered more potential threats when compared to the STRIDE threat model.

3.4 OVERALL RESULTS

When comparing the threat models PASTA is just as affective at picking up all the vulnerabilities from STRIDE due to its attack analyses step, as this allows the threats to be looked at in depth reveal multiple different ways the threat could be executed. However due to PASTA looking at common threats within the industry alongside defining the technologies that will be used it allows PASTA to pick up on more unique threats compared to STRIDE which bases its findings off the dataflow diagrams.

PASTA also manages to pick up on more practical threats compared to STRIDE such as ones involving a human element like social engineering as these threats cannot be picked up based off the dataflow diagrams alone. However, STRIDE was able to pick up on some practical threats during the repudiation stage and when it came to physical tampering.

Both the shopping and social media web applications both feature the exact same core threats as well as unique ones discovered during the research stage of PASTA. This is most likely due to both products being web applications and therefore facing the same types of threats. The dataflow diagrams created also use similar elements due to them being extremely basic versions of these types of web application, this may also play a role in the threats being similar.

Overall PASTA manages to pick up on more threats compared to using STRIDE.

4 DISCUSSION

4.1 GENERAL DISCUSSION

Due to the first two products examined being web applications they both encounter the same types of threats such as SQL injection, cross site scripting and more just within different areas of the applications, this can be seen in the threat models in sections 2.5 & 2.6. Therefore, both web applications share the same attack tree as the threat and the approaches to the threat are consistent in both. Due to these two products both being web application the results found could be inaccurate, more applications and software could be threat modeled to gain more accurate results.

STRIDE creates a straightforward threat model, allowing for anyone that views the model alongside the corresponding dataflow diagram to know exactly where threats could occur and what that potential threat could mean. Due to this method, mitigations may also be easier to implement as the areas the threats could occur are specialized. By splitting threats into categories, it makes the threat model easy to understand even for non-security specialists.

PASTA is an excellent example of threat modeling, as it allows for more than just security practitioners to participate in developing the threat model. By having to define the business objectives and technological scope various parts of a company could get together to develop a threat model, such as executives to help layout the objectives. This would potentially allow for non-security specialists to get a better idea of why cyber security is important and how common potential threats are.

4.2 ATTACK TREES

While attack trees were used within the PASTA methodology due to it being a key component during step six, they can also be used in collaboration with STRIDE. By using attack trees alongside the STRIDE methodology more potential threats may be able to be found, due to looking at multiple ways an attack can be performed.

Attack trees greatly helped in discovering various potential attacks when using the PASTA methodology. While they are an effective method of discovering attacks, they work best when used in collaboration with other threat modeling techniques.

4.3 CONCLUSIONS

Overall, both STRIDE and PASTA are extremely effective at discovering threats, with both discovering near identical potential attacks. However, while the attacks found may be the same this is because two of the products threat modeled were web applications.

During step five of the PASTA methodology more unique threats could be discovered due to this step being purely research based. This allows for the specific technologies to be investigated getting an insight

into threats only faced by those products. This was the main deviation between STRIDE and PASTA, allowing for new threats to be discovered.

Due to PASTA defining the technologies and investigating source code more potential threats could be discovered using this method. However due to this taking place before development no source code could be analyzed. The technologies used were also theoretical and once again due to this taking place before development actual threats could not be identified.

Overall, based off the findings of this report, PASTA is more affective at finding threats compared to STRIDE, due to the research phase of PASTA and the attack tree technique. However more testing should be conducted using more varied products and at different development stages, to get more data to compare.

The overall aims of the project have been met as all three products had two complete threat models created using both the STRIDE and PASTA methodologies, with the results of the comparison being PASTA is overall more affective than STRIDE but more data should be gathered to give a greater comparison.

4.4 FUTURE WORK

If this project were to be continued, in the future various ideas could be implemented. For example, an automated threat modeling tool could be created, this automated threat model process could then be compared to various manual threat modelling methodologies such as STRIDE and PASTA. By comparing the automated tool to a manual method, it could determine how viable automated threat modeling would be for the industry.

Additionally, if more time was assigned to this project the STRIDE methodology could be expanded on by combining it with attack trees. In doing this, threat's discovered by STRIDE could be further elaborated on as well as allow for additional potential attacks to be discovered, as attack trees help show multiple approaches to a singular threat outcome.

Furthermore, if this project could be done again a team could be set up to allow for potential threats to be discussed. This may allow for further threats to be discovered or different approaches to threats to be examined.

Additional products could also be tested such as media playing software and mobile applications. In doing this it would allow for more potential threats to be discovered by methodology, this would be particularly useful for STRIDE due to both web applications having near identical threat models due to facing similar threats.

Finally, if this project was taken further forward a real product could be used. This would also allow for a different approach to be taken. For example two threat models could be made for this product before development and then developed accordingly, this product can then be reviewed post launch in order to view how accurate each threat model was if it prevented anything and what they missed. Using a real product would also allow for more accurate dataflow diagrams, business objectives and technological scope.

5 REFERENCES

- Boehm, E., 2020. *Top 10 IoT Vulnerabilities in Your Devices*. [Online]
Available at: <https://blog.keyfactor.com/top-iot-vulnerabilities#lack-of-physical-hardening-10>
[Accessed 5 May 2021].
- Cynance, 2020. *PASTA threat modelling – the complete cyber security meal*. [Online]
Available at: <https://www.cynance.co/pasta-threat-modelling/>
[Accessed 21 April 2021].
- Diagrams.net, 2021. *Diagrams.net*. [Online]
Available at: <https://www.diagrams.net/>
[Accessed 18 April 2021].
- EC-Council, 2020. *6 OF THE MOST POPULAR THREAT MODELING METHODOLOGIES*. [Online]
Available at: <https://blog.eccouncil.org/6-of-the-most-popular-threat-modeling-methodologies/>
[Accessed 20 April 2021].
- Gonzalez, C., 2020. *6 Threat Modeling Methodologies: Prioritize and Mitigate Threats*. [Online]
Available at: <https://www.exabeam.com/information-security/threat-modeling/>
[Accessed 20 April 2021].
- Lucidchart, 2021. *Data Flow Diagram Symbols*. [Online]
Available at: <https://www.lucidchart.com/pages/data-flow-diagram/data-flow-diagram-symbols#:~:text=All%20data%20flow%20diagrams%20include,data%20store%20and%20data%20flow.>
[Accessed 15 April 2021].
- Microsoft, 2013. *Elevation of Privilege (EoP) Threat Modeling Card Game*. [Online]
Available at: <https://www.microsoft.com/en-gb/download/details.aspx?id=20303>
[Accessed 25 April 2021].
- Newberry, C., 2020. *Social Media Security Tips and Tools to Mitigate Risks*. [Online]
Available at: <https://blog.hootsuite.com/social-media-security-for-business/>
[Accessed 6 May 2021].
- OWASP, 2018. *OWASP IoT Top 10*. [Online]
Available at: <https://owasp.org/www-pdf-archive/OWASP-IoT-Top-10-2018-final.pdf>
[Accessed 5 May 2021].
- OWASP, 2021. *OWASP Cornucopia*. [Online]
Available at: [OWASP Cornucopia](#)
[Accessed 25 April 2021].
- OWASP, 2021. *Repudiation Attack*. [Online]
Available at: https://owasp.org/www-community/attacks/Repudiation_Attack
[Accessed 16 April 2021].

OWASP, 2021. *Threat modeling*. [Online]

Available at: https://owasp.org/www-community/Threat_Modeling

[Accessed 20 April 2021].

Poston, H., 2021. *Threat modeling tutorial: What to know before you begin*. [Online]

Available at: <https://resources.infosecinstitute.com/topic/threat-modeling-technical-walkthrough-and-tutorial/>

[Accessed 5 March 2021].

Razzak, S., 2020. *Ecommerce Security and Protection Plan for Your Online Store (2021)*. [Online]

Available at: <https://www.cloudways.com/blog/ecommerce-security-tips/#bots>

[Accessed 2 May 2021].

Schneier, B., 1999. *Attack Trees*. [Online]

Available at: https://www.schneier.com/academic/archives/1999/12/attack_trees.html

[Accessed 27 April 2021].

Shostack, A., 2014. Threat Modeling Designing for SEcurity. In: *Threat Modeling Designing for SEcurity*. Indianapolis, Indiana: Wiley, p. 627.

Simplilearn, 2021. *What is Threat Modeling: Process and Methodologies*. [Online]

Available at: <https://www.simplilearn.com/what-is-threat-modeling-article>

[Accessed 15 April 2021].

Trend Micro, 2020. *Smart Yet Flawed: IoT Device Vulnerabilities Explained*. [Online]

Available at: <https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/smart-yet-flawed-iot-device-vulnerabilities-explained>

[Accessed 5 May 2021].

Truth, S., 2011. *Create a threat model = Step 3*. [Online]

Available at: <https://blog.securityinnovation.com/blog/2011/02/create-a-threat-model-step-3.html>

[Accessed 15 April 2021].

UcedaVelez, T., 2012. *Real World Threat Modeling Using the PASTA Methodology*. [Online]

Available at: https://owasp.org/www-pdf-archive/AppSecEU2012_PASTA.pdf

[Accessed 21 April 2021].

Varghese, J., 2021. *10 E-commerce Security Threats That Are Getting Stronger By The Day!*. [Online]

Available at: <https://www.getastra.com/blog/knowledge-base/ecommerce-security-threats/>

[Accessed 4 May 2021].

VERSPRITE, 2021. *Application Threat Modeling*. [Online]

Available at: <https://versprite.com/security-offerings/appsec/application-threat-modeling/>

[Accessed 21 April 2021].

Wadhwa, M., 2020. *A Beginners Guide To The STRIDE Security Threat Model*. [Online]

Available at: https://www.ockam.io/learn/blog/introduction_to_STRIDE_security_model

[Accessed 26 April 2021].