Hochschule Offenburg
offenburg.university

ENTERPRISE AND IT COMPUTING (ENITS)
OFFENBURG UNIVERSITY OF APPLIED SCIENCES

SOFTWARE SECURITY

WINTER SEMESTER 2022/2023

THREAT MODELLING REPORT

LECTURER : SCHAAD, ANDREAS

PROF. DR. PIL. M.SC

AUTHOR :

ARINN DANISH BIN ABDULLAH (191050-01)

TABLE OF CONTENT

DOCUMENT CONVENTIONS

The following conventions are applied in this document:

- PMS: Password Management System
- LA: Legacy App
- SA: System Admin

STRIDE BACKGROUND

Praerit Garg and Loren Kohnfelder at Microsoft created the STRIDE model for identifying cyberthreats where it offers a mnemonic for each of the six categories of security concerns whereas:

- Spoofing
- Tampering
- Repudiation
- Information disclosure (privacy breach or data leak)
- Denial of service
- Elevation of privilege

STRIDE was firstly built as one part of threat modelling. In essence, STRIDE itself is a model of threat where it will be used to identify threats to one system and It is used in conjunction with a model of the target system that can be constructed in parallel. A complete overview of processes, data stores, data flows, and trust boundaries is included to it.

STRIDE is a methodological approach to assist the security experts to discover answers to "What can go wrong in this system?". Hence, each threat stated in S-T-R-I-D-E is a breach of a desired property of the system:

1) **Spoofing**

   When online thieves impersonate another person or source of information via trickery. That person is capable of manipulating modern technology, including email services and communications or the internet's core protocols.

   Example: To conceal their genuine identity in email spoofing, the adversary can compromise an unprotected mail server.

   Breach: Authenticity

2) **Tampering**

   A deliberate but unpermitted act that modifies a system, a system's component, a system's intended behaviour, or data.

   Example: Modifying the parameters in form fields. User choices are typically saved as form field values and submitted to the Web application as an HTTP request when they are made on an HTML page. These values can be hidden, free text, or pre-selected.

   Breach: Integrity

3) **Repudiation**

When an application or system doesn't have measures to accurately track and log users' actions, malicious modification or fabricating the identification of new actions might occur, leading to a repudiation attack.

Example: In a system that is unable to track the forbidden operations, a user engages in an illegal operation.

Breach: Non-repudiation

4) **Information disclosure**

When a website mistakenly exposes sensitive information available to its visitors and websites may reveal a variety of information to a potential attacker depending on the situation, including: Information about other users, such as usernames or financial information.

Example: Using a robots.txt file or directory listing to reveal the names of hidden directories, their structure, and their contents

Breach: Confidentiality

5) **Denial of Service**

A cyber-attack in which hackers or cybercriminals try to prevent the intended users from accessing a host machine, an online service, or a network resource.

Example: Numerous consumers clamouring for a deal during Black Friday discounts frequently result in a denial of service. In this scenario, an attacker deliberately tries to deplete the site's resources, preventing access for authorized users.

Breach: Availability

6) **Elevation of Privilege**

When an application acquires privileges or powers that are not appropriate for it, this is known as an elevation of privilege. Many of the threats' exploits have similarities to the elevation-of-privilege attacks.

Example: Buffer overrun attacks that cleverly attempt to write executable code.

Breach: Authorization

METHODICAL APPROACH

This report's goal is to break down a methodical strategy for using Microsoft STRIDE's threat modelling framework to analyse the Password Management System's three primary actors—user, SA, and LA in great detail. By studying a data flow diagram (DFD), keeping a lack of security controls in mind, and considering how the presence of a vulnerability in a crucial ingress point for databases, such as SQL Injection, can impact the integrity of the security of the entire system, this will further investigate the various types of threats and vulnerabilities inherent to PMS. The goal of threat modelling is to take into account various risk management strategies for reducing the threats and vulnerabilities found during the threat modelling process at the PMS design stage.

To give a systematic study of the probable attacker's profile, the most likely attack pathways, and high-value targets within the system, the process of threat modelling includes identifying, listing, and ranking potential threats and vulnerabilities against a system.

Three key questions can be answered by running a threat model of the PMS systems:

- What are the target system's assets?
- Where exactly are the system's weaknesses?
- What threats and weaknesses are most likely to be exploited?

In this report, I will be performing a threat modelling against the PMS.

Firstly, establish an asset register. Understanding the assets within the PMS is necessary before we can comprehend the dangers and weaknesses therein. The system is being broken down into its constituent parts throughout this process. Processes and system components that communicate with one another internally as well as components that external components or internal components communicate with should be included in the assets. Ingress points into OS (Operating System) operations, data stores, data flows, and trust boundaries should also be included in the asset register. We are essentially mapping out all of the parts and how they interact.

For example:

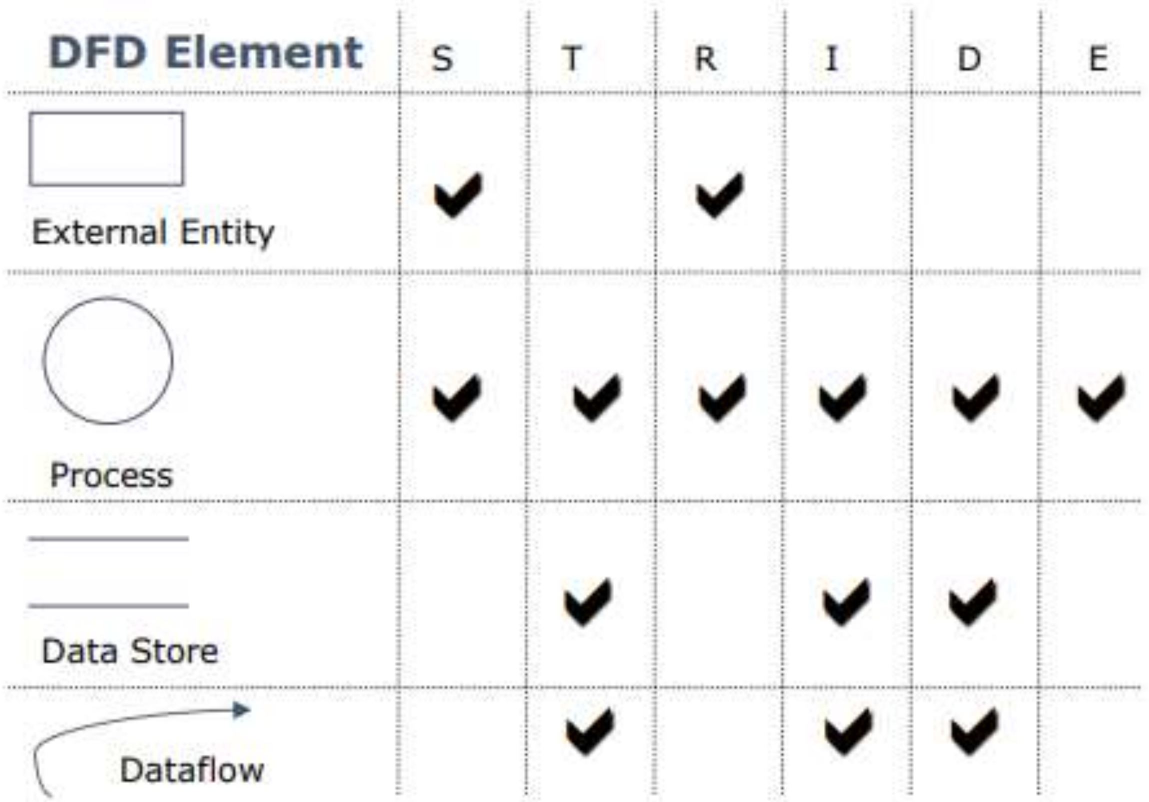
- User
- SA
- LA
- PMS
- Email
- Authorization Provider

- SQL Database

Secondly. determine the threats. The next step is to create a data flow diagram (DFD), which shows the inputs and outputs of a system's processing, transmission, and storage of data. Standard shapes are used in a DFD to depict the various components of a data flow diagram:

- External Entity (EE) (E.g. Communication endpoints)
- Process (P) (E.g. Function)
- Data Flow (DF) (Transmission of data between two elements)
- Data Storage (DS) is the storage of data, which includes files, databases, logs, data output, and trust boundaries.

Each DFD element type is in line with the STRIDE framework in the table below:



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

Diagram: Mapping of DFD elements to the STRIDE Framework

Before UML diagrams were developed, in my opinion, DFDs were used to visualize software systems. A data flow diagram specifically shows how data is transmitted between two components, known as inputs and outputs. A DFD maps the transmission, processing, and storage of data between elements in a diagram, as its name suggests.

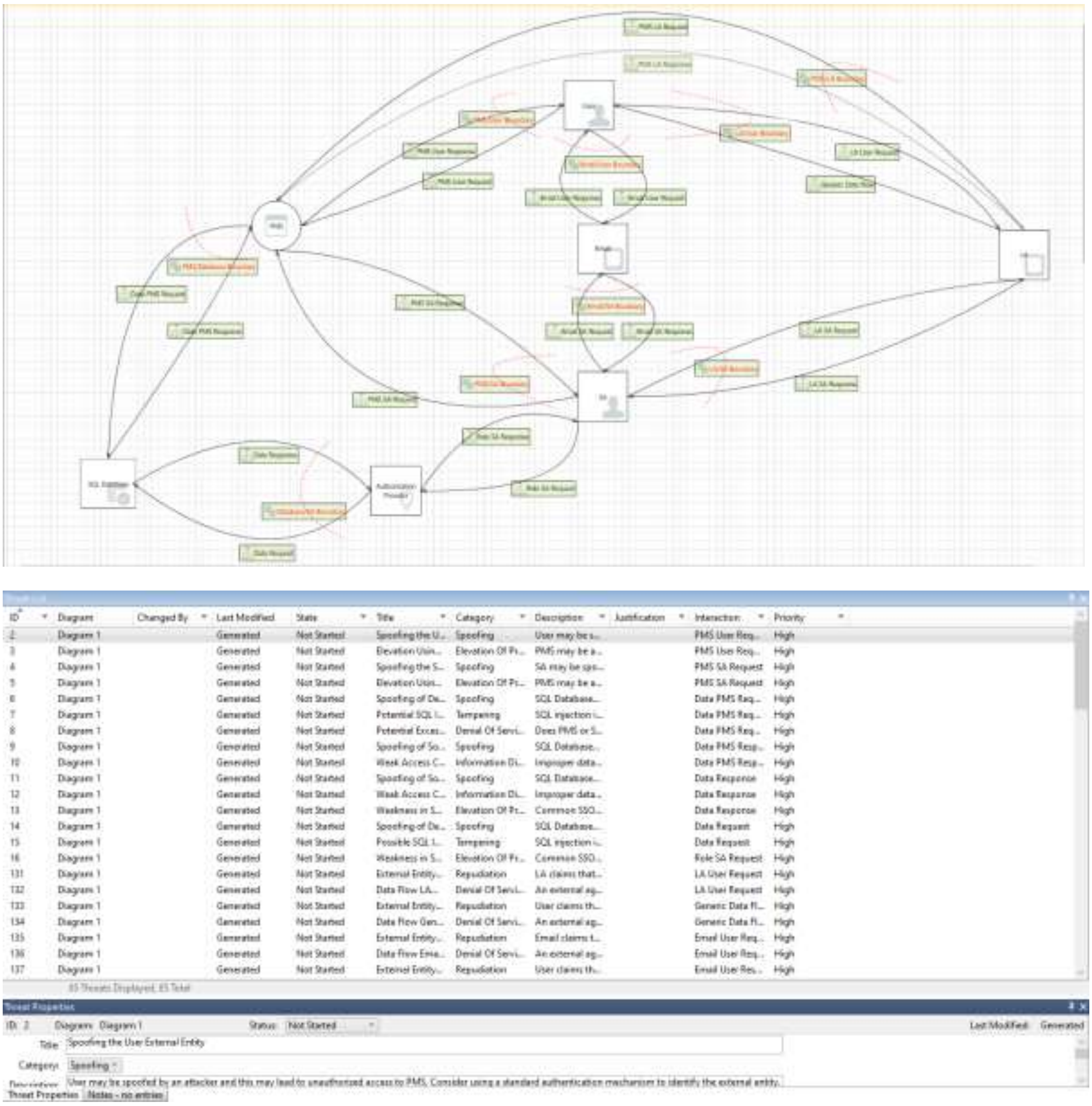
According to the STRIDE threats that were previously specified for each component, I will identify threats to the PMS in this stage. But first I have to decide how to go about doing that. For STRIDE threat modelling, there are two methodologies:

- **STRIDE-per-element**: This threat modelling technique analyses each and every unique component, which takes a lot more time. However, there are some circumstances in which a per-element approach is justified. However, it is ineffective in detecting threats that occur from component interactions. For instance, a user-to-user Email Phishing attack will only become a threat if the attacker decides to deactivate accounts or insert malware inside the email, as stated in the previous report.
- **STRIDE-per-interaction**: By taking into account the tuples (origin, destination, and interaction) of the data in transit, this sort of model counts the risks posed by interactions between components. Due to the smaller number of components that need to be modelled, this sort of modelling is far less time consuming. Due to the fact that breaches frequently entail communication between a source and destination, this kind of paradigm is excellent for cybersecurity concerns.

Thirdly, identify the vulnerabilities. Determine which of the specific assets listed in step one is vulnerable. Both passive and active vulnerability analysis would be part of this. Active analysis, for instance, involves testing traffic packets directly with endpoints like LA and analysing their responses to detect vulnerabilities rather than passive, not forwarding any packets to the destination system.

Then, the vulnerabilities will be categorized using the STRIDE approach, as described in the section above.

THREAT MODEL TOOLS



ID	Diagram	Changed By	Last Modified	State	Title	Category	Description	Justification	Interaction	Priority
2	Diagram 1		Generated	Not Started	Spoofing the U...	Spoofing	User may be s...		PMS User Req...	High
3	Diagram 1		Generated	Not Started	Elevation Usin...	Elevation Of Pr...	PMS may be a...		PMS User Req...	High
4	Diagram 1		Generated	Not Started	Spoofing the S...	Spoofing	SA may be spo...		PMS SA Request	High
5	Diagram 1		Generated	Not Started	Elevation Usin...	Elevation Of Pr...	PMS may be a...		PMS SA Request	High
6	Diagram 1		Generated	Not Started	Spoofing of De...	Spoofing	SQL Database...		Data PMS Req...	High
7	Diagram 1		Generated	Not Started	Potential SQL I...	Tampering	SQL injection i...		Data PMS Req...	High
8	Diagram 1		Generated	Not Started	Potential Exces...	Denial Of Servi...	Does PMS or S...		Data PMS Req...	High
9	Diagram 1		Generated	Not Started	Spoofing of So...	Spoofing	SQL Database...		Data PMS Resp...	High
10	Diagram 1		Generated	Not Started	Weak Access C...	Information Di...	Improper data...		Data PMS Resp...	High
11	Diagram 1		Generated	Not Started	Spoofing of So...	Spoofing	SQL Database...		Data Response	High
12	Diagram 1		Generated	Not Started	Weak Access C...	Information Di...	Improper data...		Data Response	High
13	Diagram 1		Generated	Not Started	Weakness in S...	Elevation Of Pr...	Common SSO...		Data Response	High
14	Diagram 1		Generated	Not Started	Spoofing of De...	Spoofing	SQL Database...		Data Request	High
15	Diagram 1		Generated	Not Started	Possible SQL I...	Tampering	SQL injection i...		Data Request	High
16	Diagram 1		Generated	Not Started	Weakness in S...	Elevation Of Pr...	Common SSO...		Role SA Request	High
131	Diagram 1		Generated	Not Started	External Entity...	Repudiation	LA claims that...		LA User Request	High
132	Diagram 1		Generated	Not Started	Data Flow LA...	Denial Of Servi...	An external ag...		LA User Request	High
133	Diagram 1		Generated	Not Started	External Entity...	Repudiation	User claims th...		Generic Data Fl...	High
134	Diagram 1		Generated	Not Started	Data Flow Gen...	Denial Of Servi...	An external ag...		Generic Data Fl...	High
135	Diagram 1		Generated	Not Started	External Entity...	Repudiation	Email claims t...		Email User Req...	High
136	Diagram 1		Generated	Not Started	Data Flow Ema...	Denial Of Servi...	An external ag...		Email User Req...	High
137	Diagram 1		Generated	Not Started	External Entity...	Repudiation	User claims th...		Email User Res...	High

85 Threats Displayed, 85 Total

Threat Properties

ID: 2 Diagram: Diagram 1 Status: Not Started Last Modified: Generated

Title: Spoofing the User External Entity

Category: Spoofing

Description: User may be spoofed by an attacker and this may lead to unauthorized access to PMS. Consider using a standard authentication mechanism to identify the external entity.

Threat Properties | Notes - no entries

Diagram: Microsoft Threat Modelling Tool STRIDE 2016

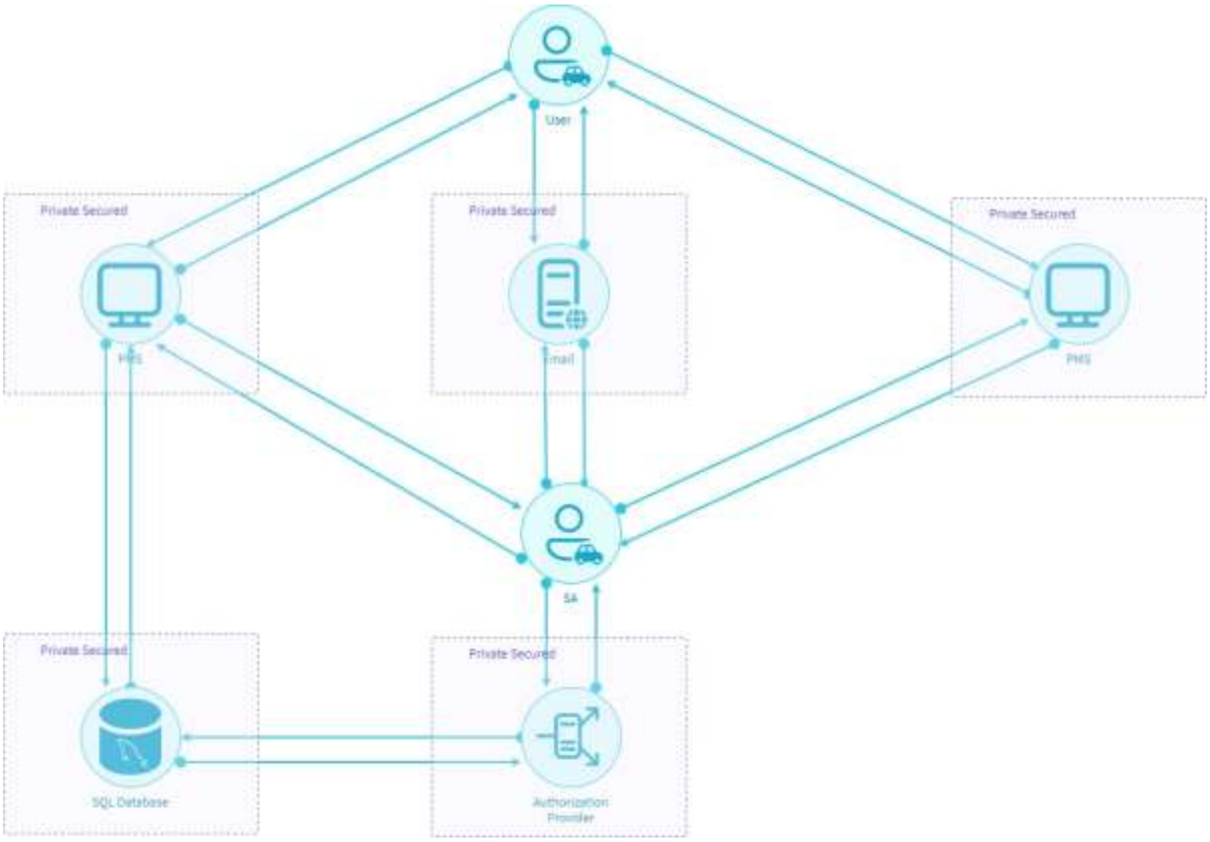


Diagram: IriusRisk Automated Threat Modelling Tool

FINDINGS & RECOMMENDATION FOR PMS

A mechanism in place that allows the users to create a batch of random passwords, set their own parameters, update password, and check for password leaks on the user's end. The system will be known as the Password Management System (PMS). Then the SA's objective to this system is quite straightforward: Blocking and unblocking user accounts, validating user accounts, editing password policies, and storing user ID and password in the PMS database are all available. The client correctly identified the business objective, but the problem description still has to be translated into specifications and strategies.

This system is clearly vulnerable to numerous dangers, many of which are implied in the issue statement's security needs. There are several issues with the password mapping process from LA to PMS alone. The most sensitive processes where attacks are most likely to occur and undermine the system as a whole include retrieving access tokens, implementing basic authentication, and storing the data in PMS databases. Identifying whether the component is a low-risk system or a high-risk component will be necessary at this point. Then, decision must be made to set out on our own, create mitigations in accordance with the classification, and learn what we don't know.

I will first make sure there isn't any data sink. Data sink refers to information that enters the analysis database but is never read (only one arrow). Because it wasn't relevant to the current issue, none of the characters or interactions that took place in the DFD explicitly referenced reading the data. It must depict the reader of the material from a security perspective. I listed out a few general guidelines to determine whether if the DFD makes sense:

1. Watch out for magical data sinks or sources: where it is crucial to ensure that an actor (e.g. user, SA and LA) is depicted as a reader or writer for each data store.
2. Ensure that a process is running at all times that can read and write data. It doesn't travel straight from a user's head to the disk, or the other way around.
3. For modelling purposes, combine similar elements inside a single trust boundary into a single element.
4. The temptation is to simulate everything concurrently on both sides of a trust border. A context DFD and more detailed breakout diagrams are good practices. The system in this case performs simultaneously simulate both the client and server systems (SQL Database).

In other words, make a better representation is crucial where the procedures for data gathering and analysis must be combined into one. Instead of implementation, consider "major function."; a simple diagram is enough for the PMS system.

Here, I will be showing the data flow that is vulnerable to the threats shown in the Microsoft STRIDE diagram above:

External Entity	Type of STRIDE	Description	Interaction
User	Spoofing	User may be spoofed by an attacker and this may lead to unauthorized access to PMS. Consider using a standard authentication mechanism to identify the external entity.	PMS User Request
	Repudiation	Email claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	Email User Request
	Information Disclosure	Data flowing across PMS User Request may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.	PMS User Request

External Entity	Type of STRIDE	Description	Interaction
SA	Elevation of Privilege	SA may be able to remotely execute code for PMS.	PMS SA Request
	Denial of Service	An external agent interrupts data flowing across a trust boundary in either direction.	Email SA Request
	Repudiation	SA claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	LA SA Response

External Entity	Type of STRIDE	Description	Interaction
LA	Denial of Service	An external agent interrupts data flowing across a trust boundary in either direction	LA User Request
	Repudiation	LA claims that it did not receive data from a process on the other side of the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.	LA SA Request

External Entity	Type of STRIDE	Description	Interaction
PMS	Denial of Service	PMS crashes, halts, stops or runs slowly; in all cases violating an availability metric	PMS LA Request
	Elevation of Privilege	An attacker may pass data into PMS in order to change the flow of program execution within PMS to the attacker's choosing.	PMS LA Request
	Information Disclosure	Data flowing across PMS LA Request may be sniffed by an attacker. Depending on what type of data an attacker can read, it may be used to attack other parts of the system or simply be a disclosure of information leading to compliance violations. Consider encrypting the data flow.	PMS LA Request

External Entity	Type of STRIDE	Description	Interaction
SQL Database	Spoofing	SQL Database may be spoofed by an attacker and this may lead to data being written to the attacker's target instead of SQL Database. Consider using a standard authentication mechanism to identify the destination data store.	Data PMS Request
	SQL Injection	SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Even parameterized data can be manipulated by a skilled and determined attacker.	Data PMS Request
	Tampering	SQL injection is an attack in which malicious code is inserted into strings	Data PMS Request

		that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Even parameterized data can be manipulated by a skilled and determined attacker.	

External Entity	Type of STRIDE	Description	Interaction
Authorization Provider	Information disclosure	Improper data protection of SQL Database can allow an attacker to read information not intended for disclosure. Review authorization settings.	Data Response
	Elevation of Privilege	Common SSO implementations such as OAUTH2 and OAUTH Wrap are vulnerable to MitM attacks.	Data response

The information displayed and created by the Microsoft STRIDE Threat Modelling Tool 2016 and the IriusRisk Automated Threat Modelling Tool is all that shown above. Now here are some recommendations and mitigations on how to prevent all STRIDE attempts from happening and recover from the attacks:

1. Spoofing

   - Implement effective antivirus protection in devices as the majority of the top free antivirus software comes with built-in tools that can identify threats in real time.
   - Change passwords frequently on PMS because if a spoofer is able to get login information, they won't be able to do anything if on fresh password.

2. Tampering

   - Encrypt sensitive data before storing I to enforce encryption for Data-at-Rest and Data-in-Transit. We will utilize encrypted connections like SSL, TLS, HTTPS, FTPS, etc. to encrypt data while it is in transit. Assign role-based controls to make sure that only authorized individuals may access the encrypted data to further tighten our data encryption. In order to improve security, we must also use multi-factor authentication.
   - Copy-on-Write File System: Delta snapshots are made each time a database is altered. Monitoring snapshots and looking for anomalous file system snapshots can help security teams find evidence of data manipulation. It becomes simpler to recover lost data, end any downtime, and restore the file system to a pre-attack condition with data in its original state.

3. Repudiation

- Cryptography, like digital signatures, is used to accomplish nonrepudiation, which also includes services for authentication, auditing, and logging. Digital signatures in PMS make sure that LA or other system actors cannot subsequently contest the delivery of information or the veracity of its signature. In public key cryptography, a digital signature is generated using the private key of an asymmetric key pair and validated using the matching public key.

4. Information Disclosure

- When feasible, use generic error messages. Avoid needlessly giving attackers hints about how the PMS and LA works.
- Verify again that any diagnostic or debugging tools are disabled in the working environment.

5. Denial of Service

- Make a DoS Response Plan. The plan's goals are to make sure that our present setup is secure, that we can identify an attack as quickly as possible, that everyone on our team is aware of their responsibilities in the event of an attack, and that the escalation and resolution processes are all understood.
- Ensure the safety of your system. Make sure the system defences are completely fortified if we want to successfully repel a DoS assault. Multi-level protection solutions using intrusion prevention and threat management systems are crucial for this. To identify and stop assaults before they overwhelm our network, these systems can use anti-spam, content filtering, VPNs, firewalls, load balancing, and security layers. Software, however, cannot do the task by itself: We will need the hardware as well.

6. Elevation of Privilege

- Check the IT infrastructure on a regular basis for any vulnerabilities that could be exploited by new threats. To do this, you must find misconfigurations, weak passwords, unpatched and insecure operating systems and applications, as well as other vulnerabilities that attackers may exploit by using a reliable vulnerability scanner.
- Employing a multi-factor authentication gives an additional layer of security while addressing potential vulnerabilities that may appear when it is challenging to manually enforce strong password regulations. Security teams should also deploy the required tools, such as policy enforcers, password auditors, and others that can scan systems, spot weak passwords, warn users, and so on. The enforcement tools make sure users have strong passwords that adhere to organizational rules in terms of length, complexity, and diversity.

PROS & CONS OF STRIDE

As security is incorporated into their development cycles, threat modelling is becoming a technology that software development teams utilize more frequently. The purpose of threat modelling was to be a highly customizable tool. Teams are allowed to choose the procedures that suit them the best while rejecting those that they think unimportant. In these years, many others questioned STRIDE, one of the procedures that has evolved over time to form a standard component of threat modelling. They were debating whether the STRIDE process was still effective for threat modelling.

In my perspective, adopting STRIDE for the less-experienced members of a threat modelling team still makes sense and is quite advantageous. The types of common attacks that today's software engineers may need to fight against are still unknown to and poorly understood by many of them. The STRIDE model might be a helpful place to start when asking these rookie threat modelers to create a list of dangers so that the team takes into account all six threat classifications.

Inexperienced threat modelers might not be aware of how an attacker might exploit exposed application-specific technical information to discover where vulnerabilities might be located within an application or feature. However, by making these inexperienced threat modelers familiar with STRIDE, they may start to see the application from the perspective of the attacker, which is a very helpful ability when trying to create more safer apps.

The downside of STRIDE, however, is that it lacks a mechanism to take into account standard frameworks (like NIST CSF, application requirements, etc.) and I observe that threat classification will in some cases been missed.

Overall, STRIDE, in my opinion, is still a useful tool for fully comprehending on what threats may our application possibly suffer in our production environment.

PROS & CONS OF USED THREAT MODEL TOOLS

Microsoft Threat Modelling Tool STRIDE 2016

Pros:

1. Easy to educate and understand, which promotes STRIDE adoption among team members who are not experts in security or technology
2. Identify high-level threats that could have an influence on the system we are modelling as soon as possible.
3. Fairly quick to complete.

Cons

1. Threats could be missed if not enough (possible) detailed threats are identified.

IriusRisk Automated Threat Modelling Tool

Pros:

1. A better, more in-depth graphical user interface to create the DFD with many more components and categories for the process, entity, data flow, and data store
2. Strong automation engine that is more scalable and collaborative than manual threat modelling, beating out its complexity.
3. A good organizational representation of DFD and reports

Cons:

1. The design environment can be too complicated for a beginner, and only we are aware of how DFD could be simplified

Overall, I find the Microsoft Threat Modelling Tool to be more user-friendly and, for a beginner, it has helped me better understand DFD and threat modelling. And even though there are certain misinterpretations regarding threat detection automation, I can easily understand the STRIDE explanation and implementation of the entire graphic.

REFERENCE

- https://en.wikipedia.org/wiki/STRIDE_(security)
- https://www.comptia.org/content/articles/what-is-spoofing#:~:text=Spoofing%20happens%20when%20cybercriminals%20use,protocols%20that%20run%20the%20internet.
- https://csrc.nist.gov/glossary/term/tampering#:~:text=Definition(s)%3A,its%20intended%20behavior%2C%20or%20data.
- https://owasp.org/www-community/attacks/Repudiation_Attack#:~:text=Description,the%20identification%20of%20new%20actions.
- https://portswigger.net/web-security/information-disclosure#:~:text=Information%20disclosure%2C%20also%20known%20as,as%20usernames%20or%20financial%20information
- https://www.checkpoint.com/cyber-hub/cyber-security/what-is-denial-of-service/
- https://www.crowdstrike.com/cybersecurity-101/spoofing-attacks/#:~:text=Spoofing%20techniques%20vary%20based%20on,activity%20and%20gather%20personal%20information.
- https://www.imperva.com/learn/application-security/parameter-tampering/#:~:text=A%20classic%20example%20of%20parameter,)%2C%20free%20text%20or%20hidden.
- https://dzone.com/articles/stride-threat-model#:~:text=Repudiation%20threats%20are%20associated%20with,to%20trace%20the%20prohibited%20operations.
- https://portswigger.net/web-security/information-disclosure#:~:text=Some%20basic%20examples%20of%20information,code%20files%20via%20temporary%20backups
- https://alissaknight.medium.com/threat-modeling-of-connected-cars-using-stride-e8184764eb0a
- https://learn.microsoft.com/en-us/archive/msdn-magazine/2006/november/uncover-security-design-flaws-using-the-stride-approach
- https://www.avast.com/c-spoofing
- https://www.cypressdatadefense.com/blog/data-tampering-prevention/
- https://www.techtarget.com/searchsecurity/definition/nonrepudiation
- https://portswigger.net/web-security/information-disclosure
- https://www.byos.io/blog/denial-of-service-attack-prevention
- https://geekflare.com/privilege-escalation-attacks/
- https://threat-modeling.com/stride-threat-modeling/#12-pros-and-cons-of-stride-threat-modeling
- https://blog.securityinnovation.com/stride