EXPERIMENT 3

Aim -

Exercise on Threat Modeling.

Theory -

What is threat modeling?
Threat modeling is a structured process with these objectives: identify security requirements, pinpoint security threats and potential vulnerabilities, quantify threat and vulnerability criticality, and prioritize remediation methods.

Threat modeling methods create these artifacts:
An abstraction of the system
Profiles of potential attackers, including their goals and methods
A catalog of threats that could arise

Methodology of Threat Modeling
1. STRIDE: STRIDE is a methodology developed by Microsoft for threat modeling. It provides a mnemonic for security threats in six categories:

- **Spoofing:** An adversary posing as another user, component, or another system that has an identity in the system being modeled.
- **Tampering:** The modification of data within the system to achieve a malicious goal.
- **Repudiation:** The ability of an adversary to deny performing some malicious activity in absence of sufficient proof.
- **Information Disclosure:** The exposure of protected data to a user that is not otherwise allowed access to that data.
- **Denial of Service:** This occurs when an adversary uses illegitimate means to assume a trust level that he currently has with different privileges.
- **Elevation of Privilege:** This threat occurs when an attacker successfully breaches the administrative controls of a system and tampers its configured permissions and privileges. By this, attackers can reach from low-level systems in the network to systems of higher authority, which contain confidential information.

2. DREAD: DREAD was proposed for threat modeling but due to inconsistent ratings, it was dropped by Microsoft in 2008. It is currently used by OpenStack and many other corporations. It provides a mnemonic for risk rating security threats using five categories. The categories are:

- **Damage Potential:** ranks the extent of damage that would occur if a vulnerability is exploited.
- **Reproducibility:** ranks how easy it is to reproduce an attack
- **Exploitability:** Assigns a number to the effort required to launch the attack.
- **Affected Users:** A value characterizing how many people will be impacted if an exploit becomes widely available.
- **Discoverability:** Measures the likelihood of how easy it is to discover the threat.

3. Process for Attack Simulation and Threat Analysis (PASTA): It is a seven-step, risk-centric methodology. The purpose is to provide a dynamic threat identification, enumeration, and scoring process. Upon completion of the threat model, security subject matter experts develop a detailed analysis of the identified threats. Finally, appropriate security controls can be enumerated. This helps developers to develop an asset-centric mitigation strategy by analyzing the attacker-centric view of an application.

Tools for Threat Modeling

1. Microsoft's Threat Modelling Tool: This tool identifies threats based on STRIDE threat model classification and is based on Data Flow Diagram (DFD), which can be used to discover threats associated with overall IT assets in an organization.

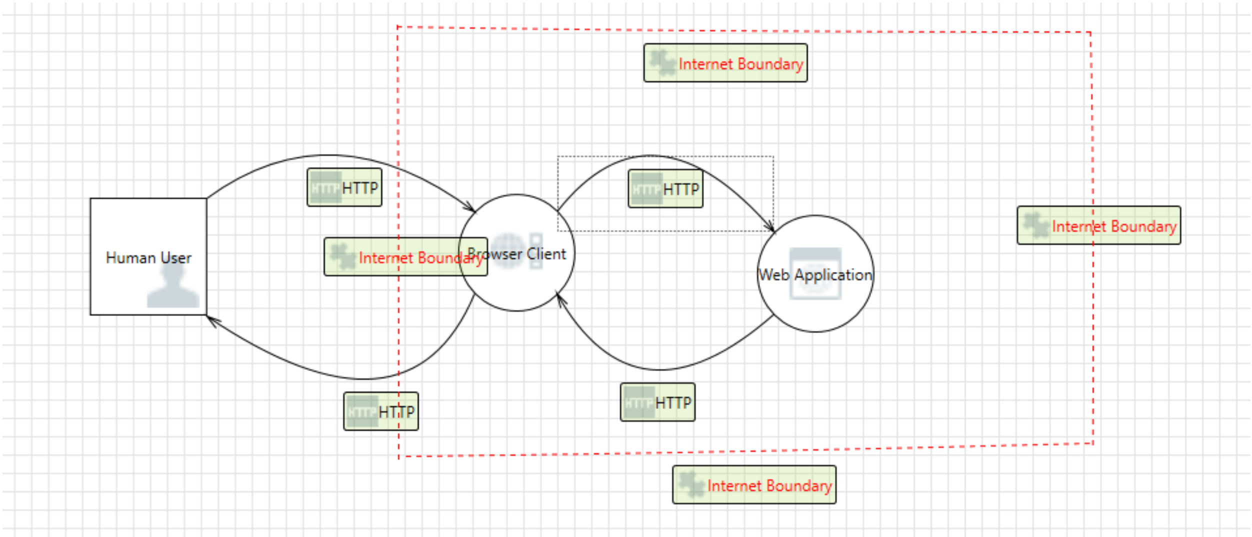
2. MyAppSecurity: It offers the first commercially available threat modeling tool – ThreatModeler. It uses a VAST threat classification scheme and it is based on a Process Flow Diagram (PFD), which provides a detailed view of the risks and vulnerable loopholes.

3. IriuRisk: Offers both a community and a commercial version of the tool. This tool is primarily used to create and maintain a live Threat model throughout the entire SDLC. It connects with several different tools like OWASP ZAP, BDD-Security, etc. to facilitate automation and involves fully customizable questionnaires and Risk Pattern Libraries.

4. securiCAD: It is a threat modeling and risk management tool developed by the Scandinavian company Foresees. Risk is identified and quantified by conducting automated attack simulations of current and future IT architectures and providing decision support based on the findings. securiCAD is offered in both commercial and community editions.

5. SD Elements by Security Compass: It is a software security requirements management platform that includes automated threat modeling capabilities. A short Questionnaire about the technical details and compliance drivers of the application is conducted to generate a set of threats. Countermeasures are included in the form of actionable tasks for developers.

Diagram -



Report-

Threat Modeling Report
Created on 7/25/2023 11:59:47 AM

Threat Model Name:

Owner:

Reviewer:

Contributors:

Description:

Assumptions:

External Dependencies:

Threat Model Summary:

Not Started	19
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	19
Total Migrated	0

Diagram: Diagram 1

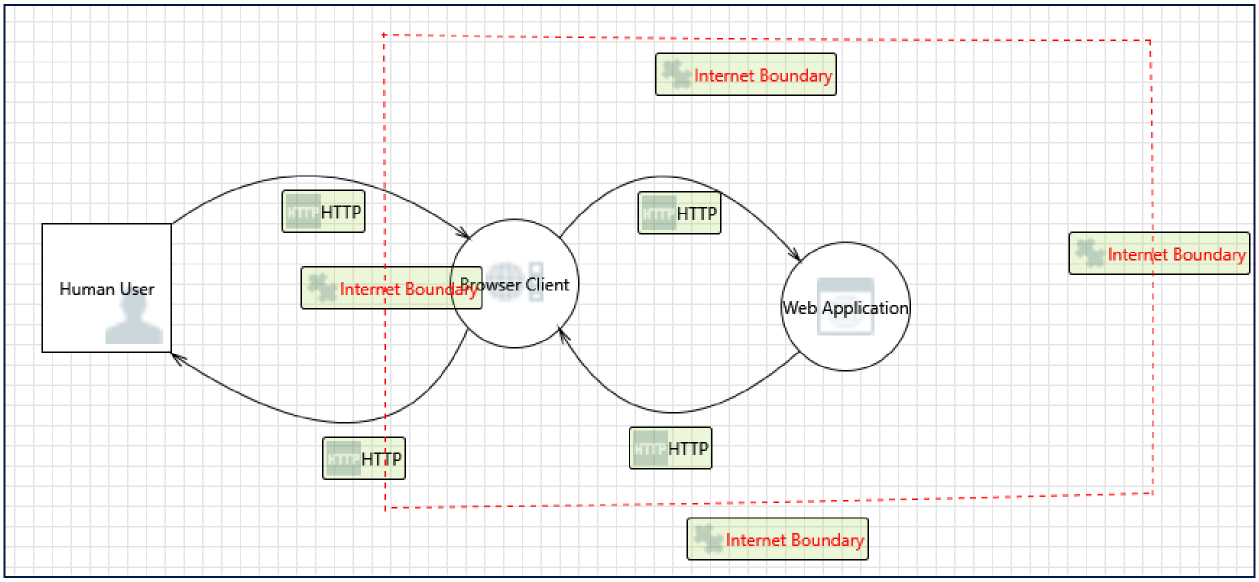
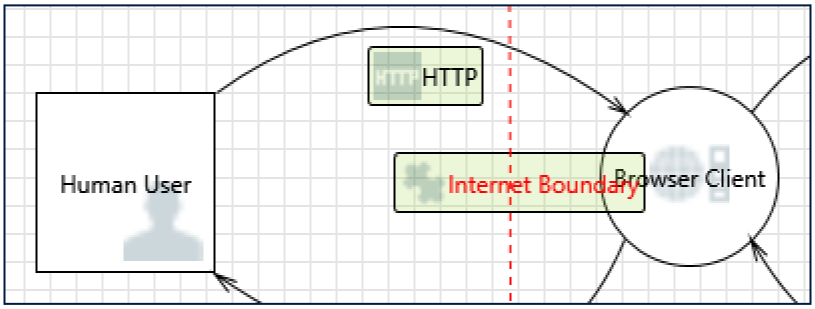


Diagram 1 Diagram Summary:

Not Started	19
Not Applicable	0
Needs Investigation	0
Mitigation Implemented	0
Total	19
Total Migrated	0

Interaction: HTTP



1. Spoofing the Human User External Entity [State: Not Started] [Priority: High]

Category: Spoofing

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

Justification: <no mitigation provided>

2. Elevation Using Impersonation [State: Not Started] [Priority: High]

Category: Elevation Of Privilege

Description: Browser Client may be able to impersonate the context of Human User in order to gain additional privilege.

Justification: <no mitigation provided>

3. Spoofing the Browser Client Process [State: Not Started] [Priority: High]

Category: Spoofing

Description: Browser Client may be spoofed by an attacker and this may lead to information disclosure by Human User. Consider using a standard authentication mechanism to identify the destination process.

Justification: <no mitigation provided>

4. Potential Lack of Input Validation for Browser Client [State: Not Started] [Priority: High]

Category: Tampering

Description: Data flowing across HTTP may be tampered with by an attacker. This may lead to a denial of service attack against Browser Client or an elevation of privilege attack against Browser Client or an information disclosure by Browser Client. Failure to verify that input is as expected is a root cause of a very large number of exploitable issues. Consider all paths and the way they handle data. Verify that all input is verified for correctness using an approved list input validation approach.

Justification: <no mitigation provided>

5. Potential Data Repudiation by Browser Client [State: Not Started] [Priority: High]

Category: Repudiation

Description: Browser Client claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.

Justification: <no mitigation provided>

6. Data Flow Sniffing [State: Not Started] [Priority: High]

Category: Information Disclosure

Description: Data flowing across HTTP 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.

Justification: <no mitigation provided>

7. Potential Process Crash or Stop for Browser Client [State: Not Started] [Priority: High]

Category: Denial Of Service

Description: Browser Client crashes, halts, stops or runs slowly; in all cases violating an availability metric.

Justification: <no mitigation provided>

8. Data Flow HTTP Is Potentially Interrupted [State: Not Started] [Priority: High]

Category: Denial Of Service

Description: An external agent interrupts data flowing across a trust boundary in either direction.

Justification: <no mitigation provided>

9. Browser Client May be Subject to Elevation of Privilege Using Remote Code Execution [State: Not Started] [Priority: High]

Category: Elevation Of Privilege

Description: Human User may be able to remotely execute code for Browser Client.

Justification: <no mitigation provided>

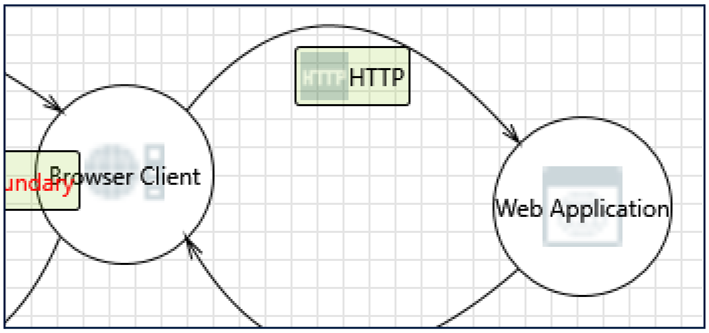
10. Elevation by Changing the Execution Flow in Browser Client [State: Not Started] [Priority: High]

Category: Elevation Of Privilege

Description: An attacker may pass data into Browser Client in order to change the flow of program execution within Browser Client to the attacker's choosing.

Justification: <no mitigation provided>

Interaction: HTTP



11. Cross Site Scripting [State: Not Started] [Priority: High]

Category: Tampering

Description: The web server 'Web Application' could be subject to a cross-site scripting attack because it does not sanitize untrusted input.

Justification: <no mitigation provided>

12. JavaScript Object Notation Processing [State: Not Started] [Priority: High]

Category: Tampering

Description: If a dataflow contains JSON, JSON processing and hijacking threats may be exploited.

Justification: <no mitigation provided>

13. Browser Client Process Memory Tampered [State: Not Started] [Priority: High]

Category: Tampering

Description: If Browser Client is given access to memory, such as shared memory or pointers, or is given the ability to control what Web Application executes (for example, passing back a function pointer.), then Browser Client can tamper with Web Application. Consider if the function could work with less access to memory, such as passing data rather than pointers. Copy in data provided, and then validate it.

Justification: <no mitigation provided>

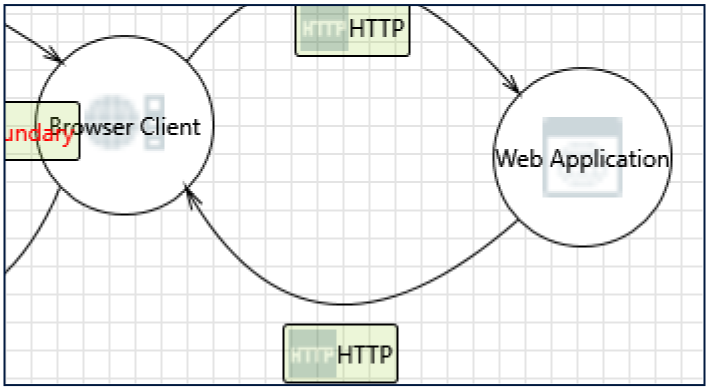
14. Elevation Using Impersonation [State: Not Started] [Priority: High]

Category: Elevation Of Privilege

Description: Web Application may be able to impersonate the context of Browser Client in order to gain additional privilege.

Justification: <no mitigation provided>

Interaction: HTTP



15. Elevation Using Impersonation [State: Not Started] [Priority: High]

Category: Elevation Of Privilege

Description: Browser Clients may be able to impersonate the context of Web Application in order to gain additional privilege.

Justification: <no mitigation provided>

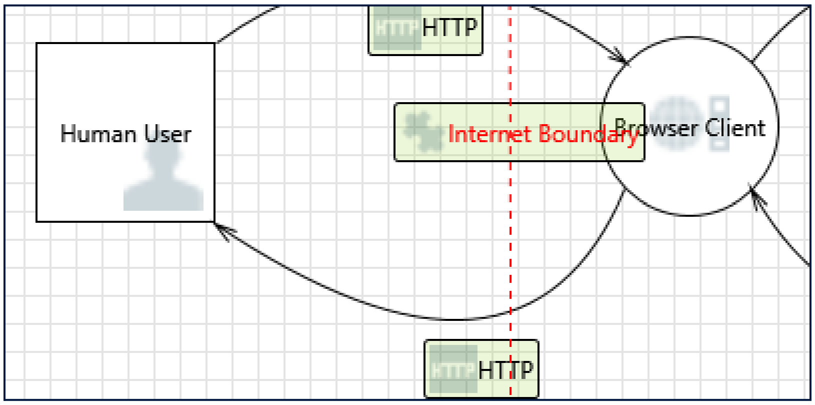
16. Web Application Process Memory Tampered [State: Not Started] [Priority: High]

Category: Tampering

Description: If Web Application is given access to memory, such as shared memory or pointers, or is given the ability to control what Browser Client executes (for example, passing back a function pointer.), then Web Application can tamper with Browser Client. Consider if the function could work with less access to memory, such as passing data rather than pointers. Copy in data provided, and then validate it.

Justification: <no mitigation provided>

Interaction: HTTP



17. Spoofing of the Human User External Destination Entity [State: Not Started] [Priority: High]

Category: Spoofing

Description: Human User may be spoofed by an attacker and this may lead to data being sent to the attacker's target instead of Human User. Consider using a standard authentication mechanism to identify the external entity.

Justification: <no mitigation provided>

18. External Entity Human User Potentially Denies Receiving Data [State: Not Started] [Priority: High]

Category: Repudiation

Description: Human User 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.

Justification: <no mitigation provided>

19. Data Flow HTTP Is Potentially Interrupted [State: Not Started] [Priority: High]

Category: Denial Of Service

Description: An external agent interrupts data flowing across a trust boundary in either direction.

Justification: <no mitigation provided>

Conclusion -

Thus we have successfully completed the experiment by identifying the issues in our application.