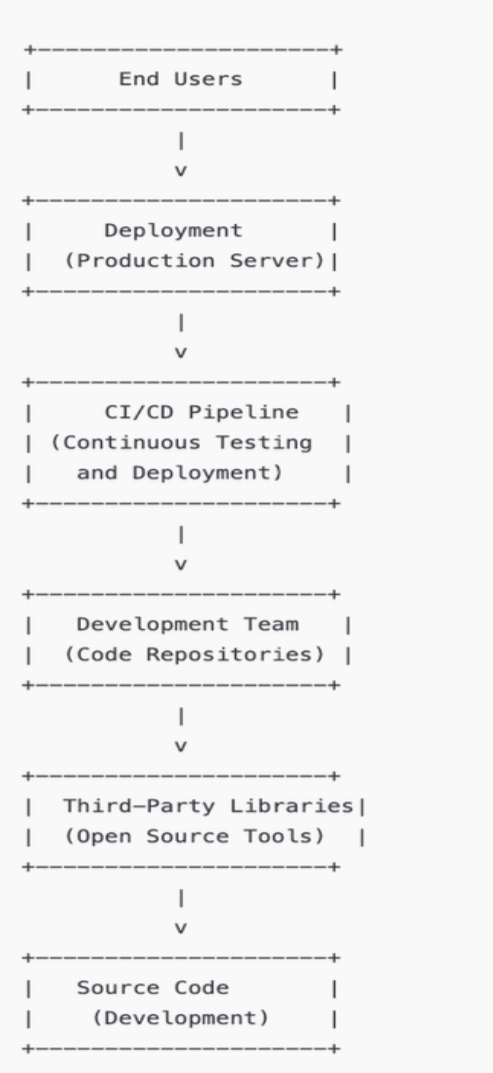
Topic 3: Risks and Threats in the Software Supply Chain

Assignment/Project/Exercise

Assignment 1: Threat Mapping Exercise

Using the sample software supply chain diagram below, students will identify potential threat points and assess their impact on the overall security of the system.



Link to the documentation site : https://cloud.google.com/software-supply-chain-security/docs/attack-vectors

Assignment 2: Case Study Analysis

Task:

Pick a real-world software supply chain attack that interests you (e.g., SolarWinds, Kaseya, Log4j, or any similar event). Dive into the details to figure out what went wrong, why it happened, and how it could have been prevented. The goal is to better understand these attacks and learn how to improve security in similar situations.

Guiding Questions:

1. **What went wrong?**
    - What specific vulnerabilities or mistakes allowed the attackers to succeed?
    - Were there any gaps in securing third-party tools or build environments that contributed to the attack?
2. **How did it happen?**
    - What steps did the attackers take to carry out the attack?
    - How did they bypass the security defenses in place?
3. **What was the impact?**
    - Who was affected by the attack, and what were the immediate and long-term consequences?
    - How did this attack expose risks to software supply chains, especially in Windows environments?
4. **What can we learn?**
    - What actions could have been taken to prevent or minimize the damage caused by the attack?
    - Suggest practical solutions and strategies to protect software supply chains from similar threats in the future.
5. **Can you visualize it?**
    - Create a diagram or flowchart showing how the attack unfolded. Highlight critical vulnerabilities or points of failure.

Deliverables:

- A **detailed report** (5-7 pages) answering the questions above, with diagrams or visuals included.
- A **15-minute presentation** where your group shares the story of the attack, your findings, and actionable recommendations to prevent similar incidents.

Solution link to the video : https://purplesec.us/breach-report/kaseya-ransomware-attack-explained/

Assignment 3: Open-source Risk Evaluation

You are tasked with performing a security risk assessment of open-source libraries integrated into a software project. Using security tools such as OWASP Dependency-Check (or other alternatives like Snyk or WhiteSource), identify known vulnerabilities within the libraries, analyze their severity, and propose mitigation strategies. If secure alternatives to the libraries are available, suggest them.

Your deliverables should include a detailed risk assessment report containing:

1. A summary of the libraries analyzed, and the vulnerabilities discovered, categorized by severity.
2. An evaluation of the potential impact and exploitability of each vulnerability.
3. Suggestions for secure alternatives or mitigation strategies for each vulnerability.
4. Recommendations for best practices to prevent the inclusion of vulnerable libraries in future projects.

Guidelines:

- Choose a set of open-source libraries used in an existing software project or select libraries from an open-source project of your choice.
- Perform a vulnerability scan and assess the risks associated with each vulnerability found.
- Provide well-supported recommendations for remediation, upgrades, or alternatives.

- Your report should be no longer than 10 pages (excluding appendices or references).

Tools/Technology Used

Tool 1: OWASP Dependency-Check

- **Purpose:** A software composition analysis tool used to identify known vulnerabilities in project dependencies.
- **Advantages:**
    - Helps ensure that third-party libraries do not introduce vulnerabilities into applications.
    - Provides real-time feedback on the security status of open-source components.
    - Facilitates automated vulnerability reporting and continuous monitoring.

Tool 2: OWASP Threat Dragon

- **Purpose:** A threat modeling tool that helps teams visualize, analyze, and address security threats within the system architecture during the design phase. It supports creating data flow diagrams (DFDs) and identifying potential vulnerabilities based on the system's design.
- **Advantages:**
    - Provides a structured approach to identifying threats and vulnerabilities early in the software development lifecycle.
    - Allows teams to model potential attack vectors and design mitigations before development starts, reducing the cost of addressing security issues later.
    - Enhances collaboration among stakeholders by providing a visual representation of potential security risks.

Sample Installation Guide:

Setting Up OWASP Threat Dragon

Step 1: Installation
Installing the Threat Dragon tool on windows desktop is a straightforward process.

- Go to the OWASP Threat Dragon page on the official OWASP website or the GitHub releases page to find the latest version of the tool.
- Under “Assets”, download the Windows installer executable file ending in .exe. For example, Threat-Dragon-ng-Setup-2.2.0.exe (Figure 4-1).

Figure 4-1. *Navigate to installer*

- Navigate to and double-click the saved .exe installer file to launch the setup wizard. You may get a security warning from Windows—click “Run” or “Yes” to allow the program to make changes to your system.

Step 2: Create a Project

Click the "Start Threat Dragon" button to begin threat modeling (Figure 4-2)!

Figure 4-2. *Start button of Threat Dragon*

Step 3: Create a Threat Model

Next is the diagramming of the model, we would require components. Figure 4-3 explains about the different stencils present and the possible components that could fall under these categories.

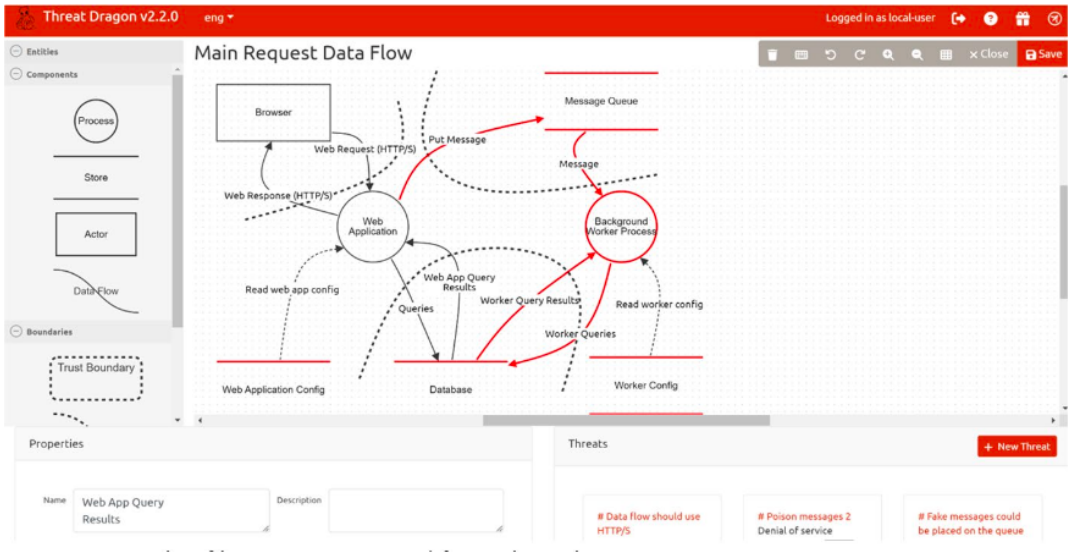


Figure 4-3. *Example of how DFD is created for web application*

An example of supply chain is how a software update package moves from development through to deployment in your supply chain. Include external entities like third-party code repositories, internal data processes like build servers, and data stores such as artifact repositories.

Step 4: Add Model Elements

- When you start Threat Dragon, click the "+" icon to create a new threat model document. Give your model a descriptive name, such as "Acme Company Web Application," to clearly define its purpose.
- On the left side, you'll find stencil palettes containing various categories of model elements like databases, users, servers, networks, and data flows. Drag these elements onto the diagram canvas to represent different parts of your system architecture. Arrange them visually to map out your environment, and double-click on any element to add text descriptions.
- Continue adding elements until you've thoroughly detailed the system you're modeling.

Step 5: Connect Model Elements

Activate the linking mode by clicking the connection icon on the left toolbar. To create connections, click on an origin element, drag your cursor to the destination element, and release. Use these connections to represent data flows, process flows, and trust boundaries, illustrating the relationships and interactions between components.

Step 6: Identify Threats

Threat Dragon automatically suggests potential threats based on the elements in your diagram. For each element, assess its security relevance by either reviewing the suggested threats or adding new ones manually.

After completing the diagram, assign threat properties to the components based on a supported framework within Threat Dragon. To do this, select a component and click the "New Threat" option to assign threats and their appropriate severity.

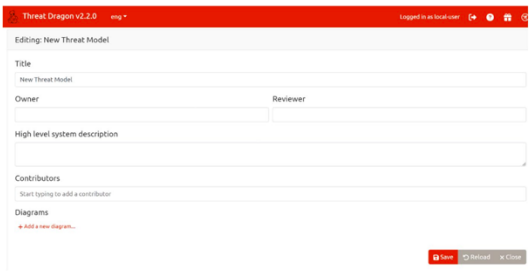


Figure 4-4. *Creating the Threat Model project*

Step 7: Report Generation

After completing the threat model, the next crucial step is generating a comprehensive report. This report provides an executive summary of the identified threats, including those mitigated and those still open, categorized by priority level. The report serves as a valuable document for reviewing the security status of your system and planning further actions to address any remaining vulnerabilities. The steps are straightforward:

- Save your model and close the page.
- You'll be redirected to the Threat Dragon home page.
- In the bottom right corner, click the "Report" option to generate your document.

Tool 3: Threat Modeling Tools (e.g., Microsoft Threat Modeling Tool)

- **Purpose:** Tools designed to visualize and analyze potential threats in the software supply chain during the design phase.
- **Advantages:**
  - Facilitates a structured approach to identifying threats and vulnerabilities in system architecture.
  - Enhances communication about security risks among stakeholders.
  - Proactively identifies potential threats early in the design phase, reducing the cost of addressing security issues later.

Tool 4: Security Information and Event Management (SIEM) Systems

- **Purpose:** Tools that provide real-time analysis of security alerts generated by applications and network hardware.

- **Advantages:**
    - Helps detect insider threats by analyzing user behavior and access patterns.
    - Provides continuous monitoring of system and network activity to identify potential security incidents.
    - Aids in incident response and forensic analysis, helping teams investigate and resolve security breaches effectively.

Interactive Activities

Activity 1: Group Discussion

- **Description:** Engage students in a collaborative discussion about the implications of recent software supply chain attacks. Students will share their thoughts on the significance of securing third-party libraries and how these incidents highlight the vulnerabilities introduced by external dependencies.
- **Objective:** The aim is to foster critical thinking and collaborative learning by encouraging students to connect theoretical knowledge with real-world scenarios. This discussion will also promote awareness of the importance of securing third-party components in the software supply chain.

Activity 2: Case Study Breakout Sessions

- **Description:** Divide the participants into small groups, each tasked with analyzing a specific software supply chain attack case study. Groups will explore the vulnerabilities exploited in the attack and collaborate to propose actionable recommendations for preventing similar incidents in the future.
- **Objective:** This activity encourages in-depth analysis and problem-solving skills by allowing students to explore real-world cases, understand the dynamics of security breaches, and come up with viable security measures. It also promotes teamwork and practical application of security concepts.

Activity 3: Hands-On Lab Session

- **Description:** Conduct a hands-on lab where students will use OWASP Dependency-Check or a similar tool to evaluate open-source libraries for known vulnerabilities. Students will practice running scans, interpreting results, and identifying potential risks in the libraries integrated into a sample project.
- **Objective:** The goal is to provide students with practical, real-world experience using industry-standard tools for securing software supply chains. This activity will help students develop technical skills in identifying vulnerabilities and securing third-party dependencies.

Assessment

Assessment Method 1: Written Test

- **Description:** A combination of multiple-choice, true/false, and short answer questions on the risks and threats in the software supply chain.
- **Objectives:** Assess knowledge retention and understanding of key concepts discussed in class.

  Sample Quiz:
  https://docs.google.com/forms/d/e/1FAIpQLScGL8yHq4_PmvCrOGN7IfxuLE-1OortSsh0U1EcTSdKFgcXZQ/viewform?usp=sharing

Assessment Method 2: Practical Assessment

- **Description:** Scenario: You are a security analyst at TechSolutions, a software development firm that creates applications for the healthcare industry. Recently, the company faced a security incident where an attacker exploited a vulnerability in a third-party library used in one of your applications. This vulnerability allowed the attacker to access sensitive patient data.

  Questions:

  1. Identify three vulnerabilities that could be associated with third-party libraries in the software supply chain.
  2. Describe the implications of each vulnerability on your application and its users, especially regarding data security and privacy.
  3. Suggest at least two strategies to mitigate these vulnerabilities effectively, focusing on how to secure the software supply chain moving forward.
- **Objectives:** Evaluate students' ability to apply theoretical knowledge to practical scenarios and suggest viable solutions.

Assessment Method 3: Group Presentation

- **Description:** Groups present their case study findings on software supply chain attacks, highlighting identified risks and countermeasures.
- **Objectives:** Assess group collaboration and the ability to communicate findings effectively.

Sample Link : https://www.youtube.com/watch?v=UcJlAoQAhS8

Title : How the Kaseya mass ransomware happened - Video on demand