**Lesson 01 Demo 01**

**Implementing DREAD Model**

**Objective:** To implement a DREAD (Damage, Reproducibility, Exploitability, Affected users, and Discoverability) model for assessing and prioritizing security risks, enabling systematic evaluation of vulnerabilities in web applications

**Tools required:** Python

**Prerequisites:** None

The **DREAD model** is a security risk assessment framework used to evaluate and prioritize vulnerabilities in web applications and other systems. It helps security teams quantify risks systematically, allowing them to address the most critical threats first.

**DREAD Components**

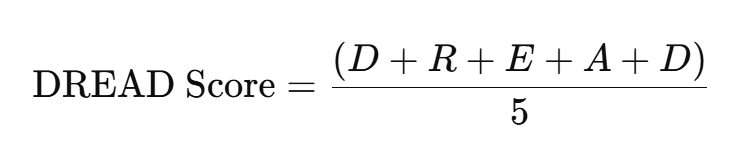
DREAD stands for:

1. **Damage Potential** – Measures the impact of an attack.
   * Example: Could the attack expose sensitive data, crash the system, or cause financial loss?
   * Rating: Low (1) to High (10).
2. **Reproducibility** – Assesses how easily the attack can be repeated.
   * Example: Can the vulnerability be exploited every time, or is it dependent on certain conditions?
   * Rating: Rare (1) to Always (10).
3. **Exploitability** – Evaluates how easy it is to exploit the vulnerability.
   * Example: Does the attack require special tools, insider knowledge, or is it trivial to execute?
   * Rating: Difficult (1) to Easy (10).
4. **Affected Users** – Estimates the number of users impacted.
   * Example: Does the issue affect only one user, a subset of users, or the entire application?
   * Rating: Few (1) to Everyone (10).
5. **Discoverability** – Determines how easy it is to find the vulnerability.
   * Example: Is it publicly documented, or would it require deep technical knowledge?
   * Rating: Hidden (1) to Obvious (10).

**Implementing the DREAD Model in Web Applications**

To use DREAD for assessing security risks in a web application:

1. **Identify vulnerabilities** – Conduct security assessments such as penetration testing, code reviews, or automated scanning.
2. **Rate each vulnerability** – Assign scores (1–10) to each factor in the DREAD model.
3. **Calculate the risk score** –



1. **Prioritize risks** – Higher scores indicate higher priority vulnerabilities that require immediate attention.
2. **Mitigate vulnerabilities** – Apply security patches, implement best practices, and continuously monitor for new threats.

**Example**

A SQL injection vulnerability is discovered in a login form. The DREAD analysis may be:

* **Damage Potential**: 9 (Leads to data exposure)
* **Reproducibility**: 10 (Can be easily repeated)
* **Exploitability**: 8 (SQL injection tools exist)
* **Affected Users**: 9 (Potentially all users)
* **Discoverability**: 7 (Moderate knowledge required)

DREAD Score:

(9+10+8+9+7)/5=8.6(9 + 10 + 8 + 9 + 7) / 5 = 8.6(9+10+8+9+7)/5=8.6

Since 8.6 is a high-risk score, this vulnerability should be fixed immediately.

**Benefits of Using DREAD**

✔ Provides a **structured** and **consistent** method for evaluating threats.  
✔ Helps teams **prioritize vulnerabilities** based on impact and exploitability.  
✔ Improves **decision-making** by focusing on the highest-risk issues.  
✔ Encourages **collaboration** between security teams, developers, and management

Steps to be followed:

1. Implement the DREAD model using a Python script

**Step 1: Implement the DREAD model using a Python script**

1. Open the **AWS Management Console** in your practice environment

A screenshot of a computer

Description automatically generated

1. Click on the cloud shell icon for the terminal  
     
   A screenshot of a computer

   Description automatically generated
2. Run the following command to verify if Python is installed:

**python3 --version**

A blue screen with white text

Description automatically generated

1. Execute the following command to create and navigate to the directory:

**mkdir python**

**cd python**  
A blue square with white lines

Description automatically generated with medium confidence

1. Run the following command to create a file:  
   **nano dread\_model.py**



* 1. Once the file opens, add the code given below, save it, and then exit:

**class DREADRiskModel:**

**def \_\_init\_\_(self, damage\_potential, reproducibility, exploitability, affected\_users, discoverability):**

**self.damage\_potential = damage\_potential**

**self.reproducibility = reproducibility**

**self.exploitability = exploitability**

**self.affected\_users = affected\_users**

**self.discoverability = discoverability**

**def calculate\_risk(self):**

**risk\_score = (self.damage\_potential + self.reproducibility +**

**self.exploitability + self.affected\_users +**

**self.discoverability)**

**return risk\_score**

**def print\_risk\_assessment(self):**

**print("DREAD Risk Model Assessment:")**

**print(f"Damage Potential: {self.damage\_potential}")**

**print(f"Reproducibility: {self.reproducibility}")**

**print(f"Exploitability: {self.exploitability}")**

**print(f"Affected Users: {self.affected\_users}")**

**print(f"Discoverability: {self.discoverability}")**

**print(f"Total Risk Score: {self.calculate\_risk()}")**

**def get\_input():**

**print("Please rate each parameter from 1 (low risk) to 10 (high risk):\n")**

**damage\_potential = int(input("Damage Potential: "))**

**reproducibility = int(input("Reproducibility: "))**

**exploitability = int(input("Exploitability: "))**

**affected\_users = int(input("Affected Users: "))**

**discoverability = int(input("Discoverability: "))**

**return damage\_potential, reproducibility, exploitability, affected\_users, discoverability**

**def main():**

**print("Welcome to the DREAD Risk Assessment Model.\n")**

**damage\_potential, reproducibility, exploitability, affected\_users, discoverability = get\_input()**

**dread\_model = DREADRiskModel(damage\_potential, reproducibility, exploitability, affected\_users, discoverability)**

**dread\_model.print\_risk\_assessment()**

**if \_\_name\_\_ == "\_\_main\_\_":**

**main()**

A screenshot of a computer program

Description automatically generated

**Note**: To save and exit the script, use CTRL + O>> press enter>> CTRL + X

* 1. Once the Python script is created, run it using the following command and follow the prompts to input ratings for the DREAD parameters as shown below:

**python3 dread\_model.py**

A computer screen shot of a computer code

Description automatically generated

The script displays the total risk score and a detailed risk assessment.

By following these steps, you have successfully implemented a DREAD model for assessing the security risks and enabling systematic evaluation of vulnerabilities in web applications.