**CVSS**

CVSS stands for COMMON VULNERABILITY SCORING SYSTEM. It’s a metric accessing the severity of the security in the software and systems. It gives a numerical value that lies between 0.00 to 10.00. These values reflect the potential impact of vulnerability.

Let’s take an example of Apache log4j JNDI Command Execution “log4shell” Vulnerability.

Step 1: Base Score Calculation

Exploitability Metrics:

* Attack Vector (AV):
  + Network (N): The vulnerability is exploitable over a network without physical access.
  + Score: 1.0
* Attack Complexity (AC):
  + Low (L): The attack does not require any special conditions or circumstances; it can be performed by anyone who can send logs.
  + Score: 0.77
* Privileges Required (PR):
  + None (N): The attacker does not need any privileges on the system to exploit the vulnerability.
  + Score: 0.85
* User Interaction (UI):
  + None (N): The attack does not require any user interaction.
  + Score: 0.85
* Scope (S):
  + Changed (C): The attack impacts not only the vulnerable component but also other components or systems within the network.
  + Score: 1.08

Impact Metrics:

* Confidentiality Impact (C):
  + High (H): The vulnerability allows an attacker to gain access to sensitive data or resources.
  + Score: 0.56
* Integrity Impact (I):
  + High (H): The vulnerability allows an attacker to modify or destroy data.
  + Score: 0.56
* Availability Impact (A):
  + High (H): The vulnerability allows an attacker to disrupt or disable the service.
  + Score: 0.56

Base Score Calculation:

The Base Score is calculated using the following formulas:

1. Impact:

Impact=1−[(1−C)×(1−I)×(1−A)]

Impact=1−[(1−0.56)×(1−0.56)×(1−0.56)]=1−(0.44×0.44×0.44)=1−0.085=0.915

1. Exploitability:

Exploitability=8.22×AV×AC×PR×UI Exploitability=8.22×1.0×0.77×0.85×0.85=4.828

1. Base Score:

* Since Scope is Changed, we use the following formula:

Base Score=min(1.08×(Impact+Exploitability),10

Base Score=min(1.08×(0.915+4.828),10)=min(6.25,10)=9.8

Step 2: Temporal Score (Optional)

Temporal metrics adjust the Base Score based on the maturity of exploit code, remediation levels, and confidence in the vulnerability report.

For this analysis, assume:

* Exploit Code Maturity (E): High (Functional exploit code is widely available) = 1.0
* Remediation Level (RL): Unavailable (No fix or mitigation available at the time of discovery) = 1.0
* Report Confidence (RC): Confirmed (The vulnerability is well-documented and confirmed) = 1.0

Temporal Score=Base Score×E×RL×RC=9.8×1.0×1.0×1.0=9.8

Step 3: Environmental Score (Optional)

Environmental metrics adjust the score based on how the vulnerability affects a specific environment. These can include factors like modified impact metrics and security requirements that vary by organization.

Final CVSS Score:

The final CVSS score for the Apache Log4j "Log4Shell" vulnerability (CVE-2021-44228) is 9.8, indicating a Critical severity level.

This score reflects the severe impact and ease of exploitation, making it a high-priority issue for organizations to address immediately.