

Vulnerabilities 101

Introduction

Cybersecurity is big business in the modern-day world. The hacks that we hear about in newspapers are from exploiting vulnerabilities. In this room, we're going to explain exactly what a vulnerability is, the types of vulnerabilities and how we can exploit these for success in our penetration testing endeavours.

An enormous part of penetration testing is knowing the skills and resources for whatever situation you face. This room is going to introduce you to some resources that are essential when researching vulnerabilities, specifically, you are going to be introduced to:

- What vulnerabilities are
- Why they're worthy of learning about
- How are vulnerabilities rated
- Databases for vulnerability research
- A showcase of how vulnerability research is used on ACKme's engagement

Introduction to Vulnerabilities

A vulnerability in cybersecurity is defined as a weakness or flaw in the design, implementation or behaviours of a system or application. An attacker can exploit these weaknesses to gain access to unauthorized information or perform unauthorized actions. The term “vulnerability” has many definitions by cybersecurity bodies. However, there is minimal variation between them all.

For example, NIST defines a vulnerability as “weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source”.

Vulnerabilities can originate from many factors, including a poor design of an application or an oversight of the intended actions from a user.

We will come on to discuss the various types of vulnerabilities in a later room. However, for now, we should know that there are arguably five main categories of vulnerabilities:

Vulnerability	Description
Operating System	These types of vulnerabilities are found within Operating Systems (OSs) and often result in privilege escalation.
(Mis)Configuration-based	These types of vulnerability stem from an incorrectly configured application or service. For example, a website exposing customer details.
Weak or Default Credentials	Applications and services that have an element of authentication will come with default credentials when installed. For example, an administrator dashboard may have the username and password of "admin". These are easy to guess by an attacker.
Application Logic	These vulnerabilities are a result of poorly designed applications. For example, poorly implemented authentication mechanisms that may result in an attacker being able to impersonate a user.
Human-Factor	Human-Factor vulnerabilities are vulnerabilities that leverage human behaviour. For example, phishing emails are designed to trick humans into believing they are legitimate.

As a cybersecurity researcher, you will be assessing applications and systems - using vulnerabilities against these targets in day-to-day life, so it is crucial to become familiar with this discovery and exploitation process.

Answer the questions below:

An attacker has been able to upgrade the permissions of their system account from "user" to "administrator". What type of vulnerability is this?

Answer: **Operating System**

You manage to bypass a login panel using cookies to authenticate. What type of vulnerability is this?

Answer: **Application Logic**

Scoring Vulnerabilities(CVSS & VPR)

Vulnerability management is the process of evaluating, categorizing and ultimately remediating threats (vulnerabilities) faced by an organization.

It is arguably impossible to patch and remedy every single vulnerability in a network or computer system and sometimes a waste of resources.

After all, only approximately 2% of vulnerabilities only ever end up being exploited (Kenna security., 2020). Instead, it is all about addressing the most dangerous vulnerabilities and reducing the likelihood of an attack vector being used to exploit a system.

This is where vulnerability scoring comes into play. Vulnerability scoring serves a vital role in vulnerability management and is used to determine the potential risk and impact a vulnerability may have on a network or computer system. For example, the popular Common Vulnerability Scoring System (CVSS) awards points to a vulnerability based upon its features, availability, and reproducibility.

Of course, as always in the world of IT, there is never just one framework or proposed idea. Let's explore two of the more common frameworks and analyze how they differ.

Common Vulnerability Scoring System

First introduced in 2005, the Common Vulnerability Scoring System (or CVSS) is a very popular framework for vulnerability scoring and has three major iterations. As it stands, the current version is CVSSv3.1 (with version 4.0 currently in draft) a score is essentially determined by some of the following factors (but many more):

1. How easy is it to exploit the vulnerability?
2. Do exploits exist for this?
3. How does this vulnerability interfere with the CIA triad?

In fact, there are so many variables that you have to use a [calculator](#) to figure out the score using this framework. A vulnerability is given a classification (out of five) depending on the score that it has been assigned. I have put the Qualitative Severity Rating Scale and their score ranges into the table below.

Rating	Score
None	0
Low	0.1 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

However, CVSS is not a magic bullet. Let's analyze some of the advantages and disadvantages of CVSS in the table below:

Advantages of CVSS	Disadvantages of CVSS
CVSS has been around for a long time.	CVSS was never designed to help prioritise vulnerabilities, instead, just assign a value of severity.
CVSS is popular in organisations.	CVSS heavily assesses vulnerabilities on an exploit being available. However, only 20% of all vulnerabilities have an exploit available (Tenable., 2020).
CVSS is a free framework to adopt and recommended by organisations such as NIST.	Vulnerabilities rarely change scoring after assessment despite the fact that new developments such as exploits may be found.

Vulnerability Priority Rating (VPR)

The VPR framework is a much more modern framework in vulnerability management - developed by Tenable, an industry solutions provider for vulnerability management. This framework is considered to be risk-driven; meaning that vulnerabilities are given a score

with a heavy focus on the risk a vulnerability poses to the organization itself, rather than factors such as impact (like with CVSS).

Unlike CVSS, VPR scoring takes into account the relevancy of a vulnerability. For example, no risk is considered regarding a vulnerability if that vulnerability does not apply to the organization (i.e. they do not use the software that is vulnerable). VPR is also considerably dynamic in its scoring, where the risk that a vulnerability may pose can change almost daily as it ages.

VPR uses a similar scoring range as CVSS, which I have also put into the table below. However, two notable differences are that VPR does not have a "None/Informational" category, and because VPR uses a different scoring method, the same vulnerability will have a different score using VPR than when using CVSS.

Rating	Score
Low	0.0 - 3.9
Medium	4.0 - 6.9
High	7.0 - 8.9
Critical	9.0 - 10.0

Let's recap some of the advantages and disadvantages of using the VPR framework in the table below.

Advantages of VPR	Disadvantages of VPR
VPR is a modern framework that is real-world.	VPR is not open-source like some other vulnerability management frameworks.
VPR considers over 150 factors when calculating risk.	VPR can only be adopted apart of a commercial platform.
VPR is risk-driven and used by organisations to help prioritise patching vulnerabilities.	VPR does not consider the CIA triad to the extent that CVSS does; meaning that risk to the confidentiality, integrity and availability of data does not play a large factor in scoring vulnerabilities when using VPR.
Scorings are not final and are very dynamic, meaning the priority a vulnerability should be given can change as the vulnerability ages.	Intentionally left blank.

Answer the questions below:

What year was the first iteration of CVSS published?

Answer: 2005

If you wanted to assess vulnerability based on the risk it poses to an organization, what framework would you use?

Note: We are looking for the acronym here.

Answer: VPR

If you wanted to use a framework that was free and open-source, what framework would that be?

Note: We are looking for the acronym here.

Answer: CVSS

Vulnerability Databases

Throughout your journey in cybersecurity, you will often come across a magnitude of different applications and services. For example, a CMS whilst they all have the same purpose, often have very different designs and behaviours (and, in turn, potentially different vulnerabilities).

Thankfully for us, there are resources on the internet that keep track of vulnerabilities for all sorts of software, operating systems and more! This room will showcase two databases that we can use to look up existing vulnerabilities for applications discovered in our infosec journey, specifically the following websites:

1. [NVD \(National Vulnerability Database\)](#)
2. [Exploit-DB](#)

Before we dive into these two resources, let's ensure that our understanding of some fundamental key terms is on the same page:


Term	Definition
Vulnerability	A vulnerability is defined as a weakness or flaw in the design, implementation or behaviours of a system or application.
Exploit	An exploit is something such as an action or behaviour that utilises a vulnerability on a system or application.
Proof of Concept (PoC)	A PoC is a technique or tool that often demonstrates the exploitation of a vulnerability.

NVD – National Vulnerability Database

The National Vulnerability Database is a website that lists all publically categorized vulnerabilities. In cybersecurity, vulnerabilities are classified under “Common Vulnerabilities and Exposures” (Or CVE for short).

These CVEs have the formatting of CVE-YEAR-IDNUMBER. For example, the vulnerability that the famous malware WannaCry used was CVE-2017-0144.

NVD allows you to see all the CVEs that have been confirmed, using filters by category and month of submission. For example, it is three days into August; there have already been 223 new CVEs submitted to this database.

[Information Technology Laboratory](#)


NATIONAL VULNERABILITY DATABASE

VULNERABILITIES

August 2021

Below is a list of CVEs for the selected month.

NOTE: The CVEs shown below have a **release date** in the year and month chosen. The CVE ID may show a year value that does not match the release date, however, the release date will fall within the chosen year and month.

223 entries found for August 2021

CVE-2021-32066	CVE-2017-18113	CVE-2021-35477	CVE-2021-34556	CVE-2021-3351	CVE-2021-24371
CVE-2021-24425	CVE-2021-24428	CVE-2021-24430	CVE-2021-24443	CVE-2021-24444	CVE-2021-24448
CVE-2021-24450	CVE-2021-24455	CVE-2021-24456	CVE-2021-24457	CVE-2021-24458	CVE-2021-24459
CVE-2021-24460	CVE-2021-24461	CVE-2021-24462	CVE-2021-24463	CVE-2021-24464	CVE-2021-24468
CVE-2021-24470	CVE-2021-24472	CVE-2021-24473	CVE-2021-24474	CVE-2021-24476	CVE-2021-24477
CVE-2021-24478	CVE-2021-24479	CVE-2021-24480	CVE-2021-24481	CVE-2021-24483	CVE-2021-24484
CVE-2021-24488	CVE-2021-24492	CVE-2021-24496	CVE-2021-24498	CVE-2021-24503	CVE-2021-24504
CVE-2021-33526	CVE-2021-33527	CVE-2021-34574	CVE-2021-34575	CVE-2021-37165	CVE-2021-37216
CVE-2021-20332	CVE-2021-37160	CVE-2021-37161	CVE-2021-37162	CVE-2021-37163	CVE-2021-37164

While this website helps keep track of new vulnerabilities, it is not great when searching for vulnerabilities for a specific application or scenario.

Exploit-DB

Exploit-DB is a resource that we, as hackers, will find much more helpful during an assessment. Exploit-DB retains exploits for software and applications stored under the name, author and version of the software or application.

We can use Exploit-DB to look for snippets of code (known as Proof of Concepts) that are used to exploit a specific vulnerability.

Date	D	A	V	Title	Type	Platform	Author
2021-08-03				Hotel Management System 1.0 - Cross-Site Scripting (XSS) Arbitrary File Upload Remote Code Execution (RCE)	WebApps	PHP	Merbin Russel
2021-08-02				Panasonic Sanyo CCTV Network Camera 2.03-0x - 'Disable Authentication / Change Password' CSRF	WebApps	Hardware	LiquidWorm
2021-08-02				Online Hotel Reservation System 1.0 - 'Multiple' Cross-site scripting (XSS)	WebApps	PHP	Mohammad Koochaki
2021-08-02				Neo4j 3.4.18 - RMI based Remote Code Execution (RCE)	Remote	Java	Christopher Ellis
2021-08-02				Men Salon Management System 1.0 - SQL Injection Authentication Bypass	WebApps	PHP	Akshay Khanna
2021-07-29				Oracle Fatwire 6.3 - Multiple Vulnerabilities	WebApps	Multiple	J. Francisco Bolivar
2021-07-29				CloverDX 5.9.0 - Cross-Site Request Forgery (CSRF) to Remote Code Execution (RCE)	WebApps	Java	niebardzo
2021-07-29				Care2x Integrated Hospital Info System 2.7 - 'Multiple' SQL Injection	WebApps	PHP	securityforeveryone.com
2021-07-29				IntelliChoice eFORCE Software Suite 2.5.9 - Username Enumeration	WebApps	ASPX	LiquidWorm
2021-07-29				Longjing Technology BEMS API 1.21 - Remote Arbitrary File Download	WebApps	Hardware	LiquidWorm
2021-07-29				Denver IP Camera SHO-110 - Unauthenticated Snapshot	WebApps	Hardware	Ivan Nikolsky

Answer the questions below:

Using NVD, how many CVEs were published in July 2021?

July 2021

Below is a list of CVEs for the selected month.

NOTE: The CVEs shown below have a **release date** in the year and month chosen. The CVE ID may show a year value that does not match the release date, however, the release date will fall within the chosen year and month.

1585 entries found for July 2021

For this I got 1585 but that wasn't the answer the question was looking for so I had to look it up and the answer they wanted is 1554.

Answer: **1554**

Who is the author of Exploit-DB?

Answer: **Offsec**

An Example of Finding a Vulnerability


In this task, I'm going to demonstrate the process of finding one minor vulnerability, coupled with some research of the vulnerability databases leading to a much more valuable vulnerability and exploit ultimately.

Throughout an assessment, you will often combine multiple vulnerabilities to get results. For example, in this task, we will leverage the "Version Disclosure" vulnerability to find out the version of an application. With this version, we can then use Exploit-DB to search for any exploits that work with that specific version.


Applications and software usually have a version number. This information is usually left with good intentions; for example, the author can support multiple versions of the software and the likes. Or sometimes, left unintentionally.

For example, in the screenshot below, we can see that the name and version number of this application is "Apache Tomcat 9.0.17"

[Home](#) [Documentation](#) [Configuration](#) [Examples](#) [Wiki](#) [Mailing Lists](#) [Find Help](#)

Apache Tomcat/9.0.17 

If you're seeing this, you've successfully installed Tomcat. Congratulations!



Recommended Reading:
[Security Considerations How-To](#)
[Manager Application How-To](#)
[Clustering/Session Replication How-To](#)

[Server Status](#)
[Manager App](#)
[Host Manager](#)

Developer Quick Start

Tomcat Setup	Realms & AAA	Examples	Servlet Specifications
First Web Application	JDBC DataSources		Tomcat Versions

Managing Tomcat

For security, access to the `manager webapp` is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 9.0 access to the manager application is split between different users.
[Read more...](#)

[Release Notes](#)
[Changelog](#)
[Migration Guide](#)
[Security Notices](#)

Documentation

[Tomcat 9.0 Documentation](#)
[Tomcat 9.0 Configuration](#)
[Tomcat Wiki](#)

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:

[Tomcat 9.0 Bug Database](#)
[Tomcat 9.0 JavaDocs](#)
[Tomcat 9.0 SVN Repository](#)

Getting Help

[FAQ and Mailing Lists](#)

The following mailing lists are available:

[tomcat-announce](#)
Important announcements, releases, security vulnerability notifications. (Low volume).

[tomcat-users](#)
User support and discussion

[taglibs-user](#)
User support and discussion for [Apache Taglibs](#)

[tomcat-dev](#)
Development mailing list, including commit messages

With this information in hand, let's use the search filter on Exploit-DB to look for any exploits that may apply to "Apache Tomcat 9.0.17".



The screenshot shows the Exploit-DB search results for the query "Tomcat 9.0". The interface includes a search bar at the top right with the text "Tomcat 9.0". Below the search bar, there are filters for "Verified" and "Has App". The results are displayed in a table with columns: Date, D, A, V, Title, Type, Platform, and Author. The table shows five entries, all of which are marked as "Verified" (indicated by a green checkmark in the 'V' column). The first entry is "Apache Tomcat 9.0.0.M1 - Cross-Site Scripting (XSS)" by Central InfoSec. The second entry is "Apache Tomcat 9.0.0.M1 - Open Redirect" by Central InfoSec. The third entry is "Tomcat proprietaryEvaluate 9.0.0.M1 - Sandbox Escape" by hantwister. The fourth entry is "Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (2)" by int0x80. The fifth entry is "Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (1)" by xxlegend. The table also shows the number of entries for each platform: Multiple (2), Java (1), JSP (1), and Windows (1).

Date	D	A	V	Title	Type	Platform	Author
2021-07-13	↓		✓	Apache Tomcat 9.0.0.M1 - Cross-Site Scripting (XSS)	WebApps	Multiple	Central InfoSec
2021-07-13	↓		✓	Apache Tomcat 9.0.0.M1 - Open Redirect	WebApps	Multiple	Central InfoSec
2020-01-08	↓		✓	Tomcat proprietaryEvaluate 9.0.0.M1 - Sandbox Escape	WebApps	Java	hantwister
2017-10-09	↓		✓	Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (2)	WebApps	JSP	int0x80
2017-09-20	↓		✓	Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (1)	WebApps	Windows	xxlegend

Great! After searching Exploit-DB, there are a total of five exploits that may be useful to us for this specific version of the application.

Answer the questions below:

What type of vulnerability did we use to find the name and version of the application in this example?

Answer: Version Disclosure

Showcase: Exploiting Ackme's Application

It is your first week on the job as Jr. Penetration tester at ThePentestingCo. For your first engagement, you are shadowing a Sr. Penetration Tester within the company.

Deploy the site attached to this task and follow the steps that the Sr. Penetration Tester took to exploit a vulnerability against ACKme IT Service's infrastructure.

Complete the engagement to retrieve a flag.

Answer the questions below:

Follow along with the showcase of exploiting ACKme's application to the end to retrieve a flag. What is this flag?

Answer: THM{ACKME_ENGAGEMENT}

Summary

Nice work! We've made it to the end. This room has served as an introductory to vulnerability research and some skills and resources this requires, where you have practically applied this knowledge.