Public Vulnerabilities

The most critical back end component vulnerabilities are those that can be attacked externally and can be leveraged to take control over the back end server without needing local access to that server (i.e., external penetration testing). These vulnerabilities are usually caused by coding mistakes made during the development of a web application's back-end components. So, there is a wide variety of vulnerability types in this area, ranging from basic vulnerabilities that can be exploited with relative ease to sophisticated vulnerabilities requiring deep knowledge of the entire web application.

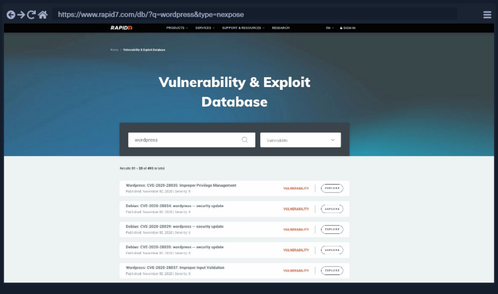
Public CVE

As many organizations deploy web applications that are publicly used, like open-source and proprietary web applications, these web applications tend to be tested by many organizations and experts around the world. This leads to frequently uncovering a large number of vulnerabilities, most of which get patched and then shared publicly and assigned a CVE (Common Vulnerabilities and Exposures) record and score.

Many penetration testers also make proof of concept exploits to test whether a certain public vulnerability can be exploited and usually make these exploits available for public use, for testing and educational purposes. This makes searching for public exploits the very first step we must go through for web applications.

> Tip: The first step is to identify the version of the web application. This can be found in many locations, like the source code of the web application. For open source web applications, we can check the repository of the web application and identify where the version number is shown (e.g., in (version.php) page), and then check the same page on our target web application to confirm.

Once we identify the web application version, we can search Google for public exploits for this version of the web application. We can also utilize online exploit databases, like Exploit DB, Rapid7 DB, or Vulnerability Lab. The following example shows a search for WordPress public exploits in Rapid7 DB:



We would usually be interested in exploits with a CVE score of 8-10 or exploits that lead to `Remote Code Execution`. Other types of public exploits should also be considered if none of the above is available.

Furthermore, these vulnerabilities are not exclusive to web applications and apply to components utilized by the web application. If a web application uses external components (e.g., a plugin), we should also search for vulnerabilities for these external components.

Common Vulnerability Scoring System (CVSS)

The Common Vulnerability Scoring System (CVSS) is an open-source industry standard for assessing the severity of security vulnerabilities. This scoring system is often used as a standard measurement for organizations and governments that need to produce accurate and consistent severity scores for their systems' vulnerabilities. This helps with the prioritization of resources and the response to a given threat.

CVSS scores are based on a formula that uses several metrics: `Base`, `Temporal`, and `Environmental`. When calculating the severity of a vulnerability using CVSS, the `Base` metrics produce a score ranging from 0 to 10, modified by applying `Temporal` and `Environmental` metrics. The National Vulnerability Database (NVD) provides CVSS scores for almost all known, publicly disclosed vulnerabilities. At this time, the NVD only provides `Base` scores based upon a given vulnerability's inherent characteristics. The current scoring systems in place are CVSS v2 and CVSS v3. There are several differences between the v2 and v3 systems, namely changes to the `Base` and `Environmental` groups to account for additional metrics. More information about the differences between the two scoring systems can be found here.

CVSS scoring ratings differently slightly between V2 and V3 as can be seen in the following tables:

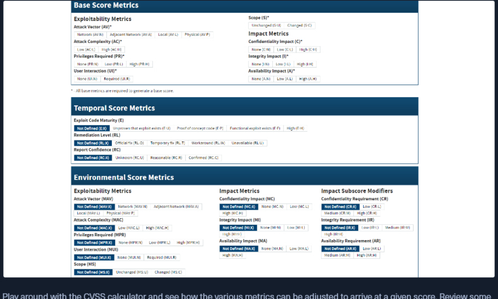
CVSS V2.0 Ratings

Severity	Base Score Range
Low	0.0-3.9
Medium	4.0-6.9
High	7.0-10.0

CVSS V3.0 Ratings

Severity	Base Score Range
None	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

The NVD does not factor in `Temporal` and `Environmental` metrics because the former can change over time due to external events. The latter is a customized metric based on the potential impact of the vulnerability on a given organization. The NVD provides a CVSS v2 calculator and a CVSS v3 calculator that organizations can use to factor additional risk from `Temporal` and `Environmental` data unique to them. The calculators are very interactive and can be used to fine-tune the CVSS score to our environment. We can move over each metric to read more about it and determine exactly how it applies to our organization. Below is an example view of the CVSS v3 calculator:



Play around with the CVSS calculator and see how the various metrics can be adjusted to arrive at a given score. Review some CVEs and attempt to arrive at the same CVSS score. How does the CVSS score change when you apply `Temporal` and `Environmental` metrics? This handy guide is extremely useful for understanding V2 and V3 and how to use the calculators to arrive at a given score.

Back-end Server Vulnerabilities

Like public vulnerabilities for web applications, we should also consider looking for vulnerabilities for other back end components, like the back end server or the webserver.

The most critical vulnerabilities for back-end components are found in web servers, as they are publicly accessible over the `TCP` protocol. An example of a well-known web server vulnerability is the `Shell-Shock`, which affected Apache web servers released during and before 2014 and utilized `HTTP` requests to gain remote control over the back-end server.

As for vulnerabilities in the back-end server or the database, they are usually utilized after gaining local access to the back-end server or back-end network, which may be gained thrown `external` vulnerabilities or during internal penetration testing. They are usually used to gain high privileged access on the back-end server or the back-end network or gain control over other servers within the same network.

Although not directly exploitable externally, these vulnerabilities are still critical and need to be patched to protect the entire web application from being compromised.

Questions

Answer the question(s) below to complete this Section and earn cubes!

+1 What is the CVSS score of the public vulnerability CVE-2017-0144?

Submit your answer here... | Submit | Hint

Previous | Next

Go to Questions

Table of Contents

Introduction to Web Applications
- Introduction
- Web Application Layout
- Front End vs. Back End

Front End Components
- HTML
- Cascading Style Sheets (CSS)
- JavaScript

Front End Vulnerabilities
- Sensitive Data Exposure
- HTML Injection
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery (CSRF)

Back End Components
- Back End Servers
- Web Servers
- Databases
- Development Frameworks & APIs

Back End Vulnerabilities
- Common Web Vulnerabilities
- Public Vulnerabilities

Next Steps
- Next Steps

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left