VULCAN.

Whitepaper

Exploit maturity: A breakdown

Table of contents

Introduction

Vulnerability scanning across multiplace attack surfaces can often yield hundreds of vulnerabilities. Of course, it's impossible to resolve all of these vulnerabilities at once, as teams lack the capacity and/or resources to do so. And with organizations under constant pressure to update and improve their network, application and cloud environments, teams must prioritize the vulnerabilities in order to know which to deal with first. This is where exploit maturity comes in handy.

Exploit maturity data enables filtering of the vulnerabilities to identify mature ones with a record of exploitation, those vulnerabilities for which there is only proof that they could be exploited, and vulnerabilities with no recorded exploitation data.

This white paper - produced by the Voyager18 research team at Vulcan Cyber® - takes an in-depth look at the exploit maturity database, so that teams can filter and prioritize system vulnerabilities quickly and accurately. Ultimately, the goal is not to fix all the vulnerabilities, but rather to fix those that could negatively impact the business.

Vulnerability maturity level

A vulnerability can be classified into one of three categories based on the exploitation records and cause:

1. Exploited
2. Proof of concept
3. Unknown

1. Exploited

A vulnerability is considered as having been exploited if there are real cases in which attackers have succeeded in exploiting the system, or if it has been verified by an author in the exploit database. An exploited vulnerability has the highest priority. Teams therefore need to remediate such vulnerabilities immediately in order to block malicious users from gaining access to sensitive data and thus prevent ransomware attacks and damage to the company's reputation.

2. Proof of concept

At this level, the exploitation has not yet been recorded, but there is a proof of concept for the exploit. The exploit, however, may be difficult to implement or it hasn't yet been used in the wild. The vulnerability can be assessed with Common Vulnerabilities and Exposures (CVE) scores. Depending on the vulnerability score (based on the Common Vulnerability Scoring System, or CVSS), the severity level can be categorized as follows:

- **Low:** 0.1–3.9 points
- **Medium:** 4.0–6.9 points
- **High:** 7.0–8.9 points
- **Critical:** 9.0–10.0 points

3. Unknown

At this level, no exploitation has been recorded yet, and there is no proof that it can be exploited. However, it is still considered to be a vulnerability in the system because a warning has been issued from the third-party libraries, the databases, or the operating systems.

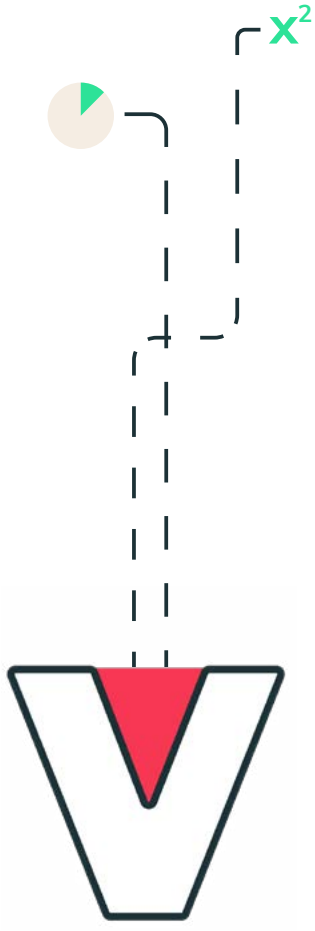
For example, the old version of an X library in Node.js uses the `Math.random()` function to generate random numbers, which is considered security sensitive since pseudo-random numbers are generated instead. The vulnerability scanning tool will suggest that the developers upgrade to the latest version of an X library that uses `Crypto.getRandomValues()`, a cryptographically strong random generator, instead.

Criteria for maturity levels

Not all vulnerabilities that are categorized as "exploited" need to be fixed immediately. There are several criteria that need to be considered so that teams can identify which vulnerabilities need to be fixed first, which can be fixed later, and which can be addressed through a workaround or temporary fix.

Following are the criteria that affect the exploit maturity levels:

1. Effort needed to make the exploitation work



First, teams need to take into account how much effort is needed to make the exploitation work. The more effort it takes, the lower the exploit maturity level of the vulnerability.

Amount of work

How much work is needed to make the exploitation work? For instance, many steps may need to be executed in order to make an exploitation work, such as:

- Registering a new account in the customer portal
- Successfully purchasing the product
- Registering the purchased product kit with the current account
- Calling an API that has a vulnerability

In this example, the exploitation will have a much lower maturity level compared to one for which all it takes to make the exploit work is to register a new account and call some weak API.

Ease of exploitation

To exploit the system using the vulnerability, do malicious users need to know about the features of internal services used in the admin portal? Do they simply need to go to the customer portal and log in as a regular user in order to exploit the vulnerability? Do attackers need access to the company network because the vulnerability cannot be exploited from the public internet? The easier it is to make the exploitation work, the higher the maturity level of the vulnerability.

2. Exploit availability

The exploit maturity level also depends on the exploit availability. If the vulnerability appears in the exploit database, it should be remediated right away. The exploit database is a well-known resource for exploitable vulnerabilities and one of the very first means attackers turn to in order to gain access to the company system.

If the exploited vulnerability is only available in particular areas, such as China or Russia, the maturity level will be lower.

3. Impact

For every exploited vulnerability, the impact on the company product and reputation can differ. For example, a vulnerability that exposes users' birthdays, will have far less impact than one that exposes users' bank account numbers. Similarly, a vulnerability allowing attackers to access the AWS bucket that stores customers' avatars has less impact than one that allows access to the bucket that stores users' billing reports.

Security analysis should therefore assess the impact an exploit would have in terms of confidentiality, integrity, or availability.

Confidentiality

Protecting users' personal data is critical, especially in the fields of e-commerce and bioinformatics. Customer data such as genetic information must be kept private, with strict authorization regulations to prevent malicious users from gaining access to sensitive information such as health records or the identity of an individual's biological parents.

Proper implementation of authentication procedures and data encryption is key to maintaining confidentiality. For example, teams should use bcrypt to encrypt user passwords and should apply two-factor authentication (2FA) in order to log in to any software application.

Integrity

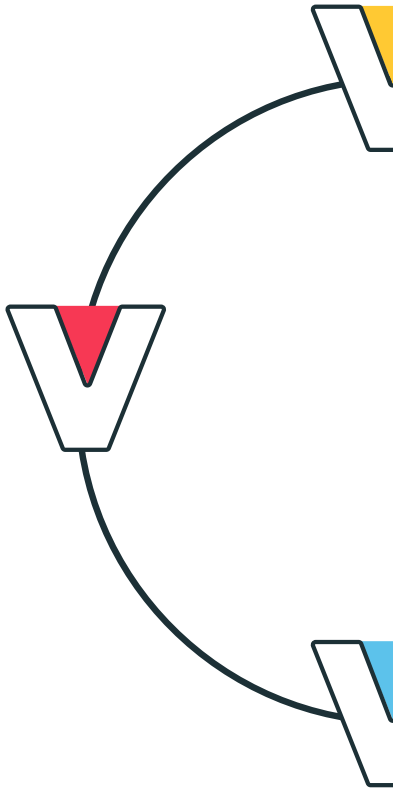
In 2016, a cyber-security attack recorded in the central bank of Bangladesh at the Federal Reserve Bank of New York generated multiple fraudulent withdrawals equivalent to around $1 billion U.S. dollars. The hackers exploited a security vulnerability in the banking system to retrieve the necessary credentials, then injected malware to delete database records of the illegal transfers.

Solutions for preventing such integrity violation issues include hash verification or digital signatures to ensure that each transaction is made by authorized users and their transactions cannot be modified or corrupted once created.

Solutions for preventing such integrity violation issues include hash verification or digital signatures to ensure that each transaction is made by authorized users and their transactions cannot be modified or corrupted once created. In addition, to achieve maximum system integrity, teams should consider using blockchain technology, which enables decentralization and ensures that members' transaction records in distributed networks are immutable.

Availability

For most software products, availability is a top priority, especially in the fields of e-commerce, banking, and social networks. Disruption of availability can lead to loss of customers and can harm company reputation.



Distributed denial-of-service (DDoS) is the most widespread cyber-security issue affecting system availability. There are three types of DDoS attacks:

- **Application-layer attack**: Also known as Layer 7 DDoS attacks, this type of attack focuses on the existing vulnerabilities in the application layer, causing greater damage to the victim's servers, but with less traffic generation.
- **Protocol attack**: Makes use of vulnerabilities in internet communication protocols to access network infrastructures such as servers, load balancers to flood the servers, or redirect original requests to other routes.
- **Volumetric attack**: In this most common type of DDoS attack, the victim's server is flooded with an extremely high number of requests, to the point that the server can no longer handle incoming requests. As a result, real users experience a timeout or internal server error when using an application.

In 2020, AWS experienced a DDoS attack unprecedented in size: attackers used the Connectionless Lightweight Directory Access Protocol (CLDAP) reflection method to target an AWS customer. They exploited a vulnerability found on third-party CLDAP servers, then amplified the amount of data sent to the victim's IP address by more than 50 times the original data.

In 2021, Akamai announced they had dealt with some of the biggest DDoS attacks to date, which targeted a European gambling company. The attacks abused protocol 33—a new DDoS attack vector—with volumetric traffic.
In order to protect company systems from DDoS attacks, several techniques should be used in combination, such as applying firewalls for an additional layer of defense or enabling the system to scale out for servers and network systems alike.

4. Scope

The scope of exploitation is another important factor. An exploitation may affect a single service or the whole company system.

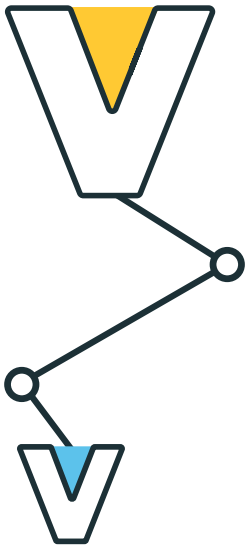
In 2017, the Google Project Zero team discovered that Cloudflare's servers were allowing sensitive data to be cached by search engines. Later, the Cloudflare team acknowledged the problem, noting it had likely started in September 2016, as the result of a leaked private key between Cloudflare servers. At that time, approximately six million websites were using Cloudflare's services, and Cloudflare affirmed that between September 2016 and February 2017, the problematic caching mechanism was triggered 1,242,071 times.

In Cloudflare's case, the scope did not stop at the application level, but affected multiple web applications from around the world.

Verifying exploit maturity

The next step is to verify whether the vulnerability could escalate and lead to severe security incidents in the future if not addressed. Ultimately, fixing a vulnerability can take significant time and effort. It may also have side effects that can disrupt the existing functionalities or even introduce additional security vulnerabilities into the system.

Remediating the vulnerability



Once the exploit has been proven, the next step is to fix the vulnerability.

The fix can be as simple as upgrading the related library to the latest version. Alternatively, the service's entire architecture may need to be redesigned in order to remediate it. The more complicated the fix, the more impact it will have on existing functionality of the company system. Security issues should thus be identified early on in order to minimize problems and reduce the costs of remediating them.

Retry the exploit maturity steps
First, the team needs to apply the solution to the test environment in which they simulated the exploitation. Then, they should retry the exploit maturity steps to determine whether the vulnerability has been completely remediated.

Check for any side effects
Once the vulnerability has been remediated, it is necessary to ensure the fix doesn't cause any new system issues. This requires conducting a thorough regression test for all functionalities. Performance tests should be executed to ensure the vulnerability remediation hasn't impacted the performance of related services. Finally, the system should be scanned to make sure there are no exploitable vulnerabilities left.

Double-check for any backdoors
Although teams use a test environment when simulating the exploitation, there might be some configurations or components that were modified for the simulation that could affect the security status of the company system. Therefore, security experts must double check whether there are any backdoors, problematic configurations, or if any sensitive data is exposed. the existing functionalities or even introduce additional security vulnerabilities into the system.

Conclusion

Efficient prioritization of vulnerability maturity level requires up-to-date exploit maturity databases. The process of checking for exploit maturity should be integrated with vulnerability scanning. This allows teams to prioritize vulnerabilities immediately after they are found. The sooner high-priority vulnerabilities are addressed, the better the chance of preventing the system from being exploited.

About Vulcan Cyber

Vulcan Cyber® breaks down organizational cyber risk into measurable, manageable processes to help security teams go beyond their scan data and actually reduce risk. With powerful prioritization, orchestration and mitigation capabilities, the Vulcan Cyber risk management SaaS platform provides clear solutions to help manage risk effectively. Vulcan Cyber enhances teams' existing cyber environments by connecting with all the tools they already use, supporting every stage of the cyber security lifecycle across cloud, IT and application attack surfaces. The unique capability of the Vulcan Cyber platform has garnered Vulcan recognition as a 2019 Gartner Cool Vendor and as a winner of a Global InfoSec Award at RSA Conference 2022.

Start owning your risk

TRY VULCAN CYBER

About Voyager18

The Vulcan Cyber research team, also known as Voyager18, is a team of cyber experts working to leverage machine-learning and cyber research to ensure Vulcan Cyber remains a cyber security leader in the field. The team's main objective is to research the latest cyber risk trends, including new attack types and remediations. The team is also responsible for bringing innovation to the Vulcan Cyber platform so that our customers get improved and customized cyber risk management capabilities. This includes research of more specific and accurate risk calculations that can truly help our customers own their risk. Most recently, the team mapped out the MITRE ATT&CK framework to relevant CVEs, providing granular insights into the most critical vulnerabilities. The full research is available here.

Stay up-to-date with latest research trends

EXPLORE MORE