

Pentest Report Example 2025

Pentest Report

Hannes Seidl

01/01/2025

- 1 Executive Summary
 - 1.1 Synopsis
 - 1.2 Findings Overview
 - 1.3 Strategic Recommendations
 - 1.3.1 Enterprise Patch Management Policy
 - 1.3.2 Tier-Model for Active Directory
 - 1.3.3 Demilitarized Zone Hardening
- 2 Test Scope
 - 2.1 External Scope
 - 2.1.1 Exclusions
 - 2.1.2 Limitations
 - 2.2 Internal Scope
 - 2.2.1 Exclusions
 - 2.2.2 Limitations
- 3 Findings
 - 3.0.1 Affected Systems
 - 3.0.2 Severity (CVSS)
 - 3.0.3 Short Description
 - 3.0.4 Technical Description
 - 3.0.5 Remediation
 - 3.0.6 References
 - 3.1 Unauthorized Access To Nextcloud
 - 3.1.1 Affected Systems
 - 3.1.2 Severity (CVSS)
 - 3.1.3 Short Description
 - 3.1.4 Technical Description
 - 3.1.5 Remediation
 - 3.1.6 References
 - 3.2 Sensitive Data Exposure on SMB Shares
 - 3.2.1 Affected Systems
 - 3.2.2 Severity (CVSS)
 - 3.2.3 Short Description
 - 3.2.4 Technical Description
 - 3.2.5 Remediation
 - 3.2.6 References
 - 3.3 Reflected XSS Vulnerability
 - 3.3.1 Affected Systems
 - 3.3.2 Severity (CVSS)
 - 3.3.3 Short Description
 - 3.3.4 Technical Description
 - 3.3.5 Remediation
 - 3.3.6 References
 - 3.4 Kerberoastable Accounts in Active Directory
 - 3.4.1 Affected Systems

- 3.4.2 Severity (CVSS)
 - 3.4.3 Short Description
 - 3.4.4 Technical Description
 - 3.4.5 Remediation
 - 3.4.6 References
- 3.5 Username Enumeration By Response Timing
 - 3.5.1 Affected Systems
 - 3.5.2 Severity (CVSS)
 - 3.5.3 Short Description
 - 3.5.4 Technical Description
 - 3.5.5 Remediation
 - 3.5.6 References
- 3.6 Insecure Content Security Policy
 - 3.6.1 Affected Systems
 - 3.6.2 Severity (CVSS)
 - 3.6.3 Short Description
 - 3.6.4 Technical Description
 - 3.6.5 Remediation
 - 3.6.6 References
- 4 Appendix
 - 4.1 CVSS Explanation
 - 4.1.1 Components of a CVSS Vector
 - 4.1.2 1. Base Metrics
- 5 Password Security Glossary
 - 5.0.1 Why This Matters
 - 5.0.2 How to Protect Yourself

1 Executive Summary

1.1 Synopsis

This is an example report for showcasing the stucture of a pentest report. Various common Findings were picked and described.

Hannes Seidl was tasked by **Example Inc.** to perform penetration tests on their network. For this test, including documentation, **5 project days** have been spent.

During the examination of the infrastructure, **2 critical severity**, **3 medium severity** and **1 low severity** vulnerabilities were discovered. Additionally, **1 informational finding** was noted.

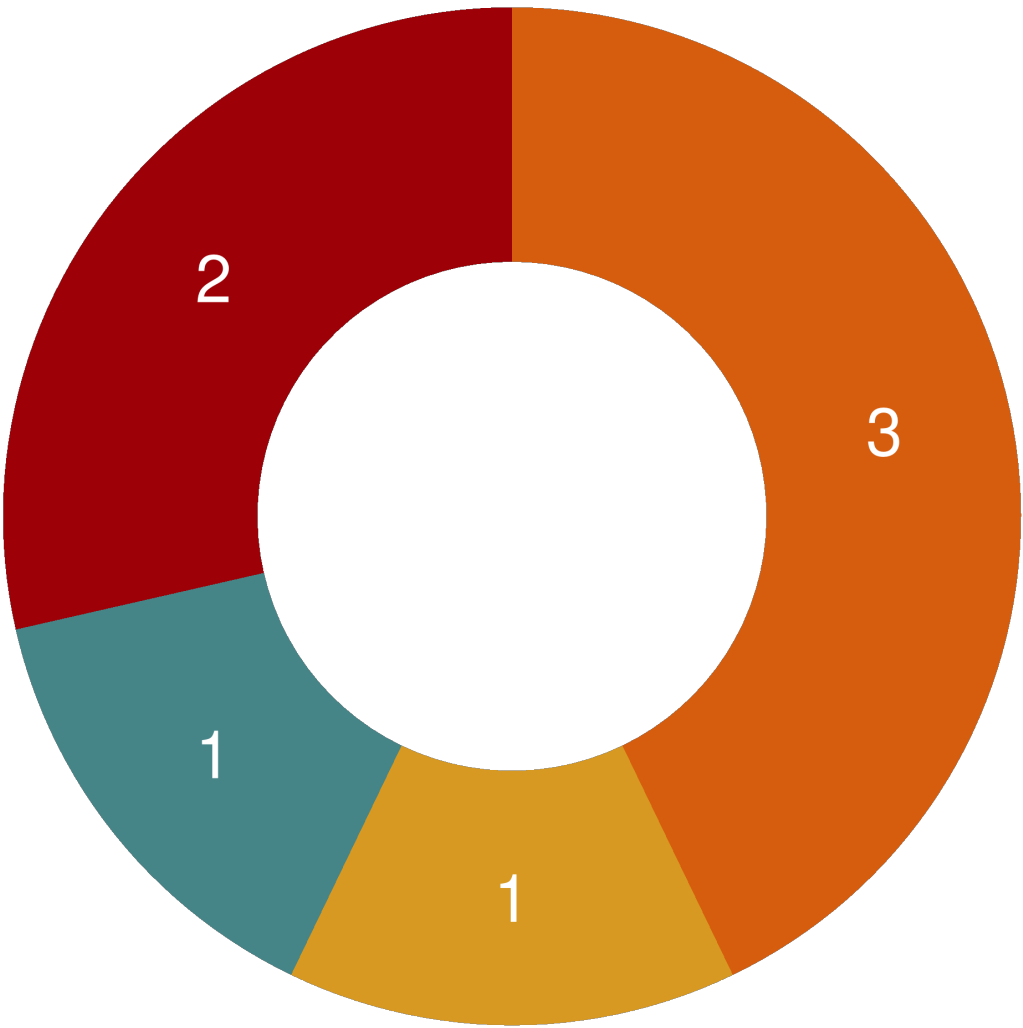
 Due to the severity of the discovered vulnerabilities, it was possible to take over critical infrastructure belonging to **Example Inc.**

We recommend a prioritized remediation of the vulnerabilities, with respect to their CVSS scores and perceived business impacts. Please note that the CVSS scores represent a purely technical evalutation of the vulnerability, and not the business

impact. Advice for remediation for each respective vulnerability is located in the Findings section.

1.2 Findings Overview

Below is an overview of all the findings from the test:



■ Critical ■ Info ■ Low ■ Medium

FINDING NAME	CVSS	SEVERITY
Default Password for Telnet Interface	10.0	CRITICAL

FINDING NAME	CVSS	SEVERITY
Unauthorized Access To Nextcloud	10.0	CRITICAL
Senstive Data Exposure on SMB Shares	6.5	MEDIUM
Reflected XSS Vulnerability	6.1	MEDIUM
Kerberoastable Accounts in Active Directory	4.1	MEDIUM
Username Enumeration By Response Timing	3.5	LOW
Insecure Content Security Policy	0.0	INFORMATIONAL

1.3 Strategic Recommendations

This section provides a high-level overview of the recommended actions for enhancing the security posture of your organization.

1.3.1 Enterprise Patch Management Policy

In order to strengthen the security posture and resilience of your IT infrastructure, it is essential to implement a robust patch management policy. This policy should be strategically designed to ensure that all systems remain protected against known vulnerabilities, minimizing the attack surface that could be exploited by adversaries. Specifically, first steps in this direction could be:

1. Use recent versions of products.
2. Ensure Windows Server version is fully patched and up to date and still supported.
3. Use a recent version of a Backup System.

Using old software with known vulnerabilities exposes your business to risk of compromise which can lead to excessive financial or reputational damages. If updating is not possible, devices should be segregated into a separate network.

1.3.2 Tier-Model for Active Directory

To introduce further hurdles for potential attackers and follow the Principle of Least Privilege, multiple tiers of Administrator accounts can be utilized. For example, a basic 3-Tier model would consist of the following:

- *Tier-0*: DC Admin
- *Tier-1*: Server Admin
- *Tier-2*: Workstation Admin

Each of these should only have sufficient privilege to access the intended resources. In this manner you can reduce the chances of an attacker to gain further privileges in the event of a system compromise, e.g. taking over a workstation does not allow the attacker to directly move to a server or vice versa.

It should be mentioned that Tier-0 accounts should ideally only be accessed via a separate workstation, such that compromise of the Domain Administrators workstation does not lead to a full domain takeover.

1.3.3 Demilitarized Zone Hardening

To keep the internal network safe from attacks, incoming connections from the DMZ should be blocked and no domain credentials should be leaked even in case of a network breach.

1. *Blocking Incoming Connections from the DMZ*: This can be achieved by configuring the firewall rules to deny all incoming traffic from the DMZ network to your internal network. Outgoing connections from the DMZ hosts to the internal network should be allowed only if absolutely necessary, and even then, they should be strictly controlled and monitored.
2. *Avoiding Leakage of Domain Credentials*: To prevent credential leaks in case of a breach, you should:
 - Use local accounts for DMZ hosts instead of domain accounts. This limits the scope of any compromised credentials to only those hosts within the DMZ and prevents an attacker from potentially gaining access to the entire internal network.
 - Implement strict password policies and enforce multi-factor authentication (MFA) where possible.
 - Regularly rotate credentials and audit permissions to ensure the principle of least privilege is maintained.

2 Test Scope

The Test Scope section defines the boundaries and parameters of the penetration test. It outlines the specific targets, such as networks, applications, and systems, that are to be evaluated. It also specifies any limitations or exclusions that were agreed upon before the test commenced. The purpose is to provide a clear understanding of what was tested and ensure that the assessment was conducted within the agreed-upon constraints.

2.1 External Scope

Following details the scope for the external testing phase:

- *.example.com

2.1.1 Exclusions

Following targets are excluded from the scope and not to be tested:

- web.prod.example.com

2.1.2 Limitations

- Denial of Service attacks

2.2 Internal Scope

Following details the scope for the internal testing phase:

- 192.168.1.0/24

2.2.1 Exclusions

- 192.168.1.233

2.2.2 Limitations

- Denial of Service attacks

3 Findings

Several key findings have emerged which highlight both vulnerabilities and areas for significant improvement. Each finding is categorized based on its impact level and poses a distinct risk to the organization's information security and operational integrity. Immediate attention to these issues is recommended to protect against potential cyber threats.

Each finding is structured in the following manner:

- **Finding Title**
- **Affected Systems:** The list of systems (IP addresses, hostnames, domain names, application or device name) which are confirmed to be affected by the issue.
- **Severity (CVSS):** The CVSS (Common Vulnerability Scoring System) score for the finding will be presented here. For more details on CVSS, see the Appendix section CVSS Explanation.
- **Short Description:** A brief description for the finding without going into the details. The impact of the vulnerability may be also mentioned here.
- **Technical Description:** This section describes the technical details of each finding. All the details about reproducing the issue will be detailed in here, such as code snippets for analysis, HTTP communication logs, screenshots which depict the problem or a successful exploitation of the vulnerability in question. The impact of the vulnerability is also discussed here with more details and example exploitation scenarios.
- **Remediation:** Advice and recommendations for a successful mitigation of the finding. The remediation steps will vary between each type of vulnerability.

It may include instructions or references to instructions to describe technical measures, as well as describe organizational improvements to prevent the reoccurrence of the vulnerability.

- **References:** Links to various resources which may aid in the understanding or for the mitigation of the vulnerability in question. ## Default Password For Telnet Interface (Command Execution)

3.0.1 Affected Systems

- router.example.local (192.168.1.1)

3.0.2 Severity (CVSS)

CVSS 3.1 Vector: 10.0 - CRITICAL

AC:L / PR:N / UI:N / S:C / C:H / I:H / A:H

3.0.3 Short Description

Default credentials allow anyone to login to the interface and execute any commands.

3.0.4 Technical Description

The host uses a known default password, with which an attacker in the same network can dial into and execute commands as root on the underlying operating system.

🔥 This leads to a full compromise of the router, which puts the security of the network traffic into question.

An attacker with this level of access can see and manipulate network traffic, such as sniffing plaintext traffic for credentials, or manipulating the responses of DNS queries to point users to malicious phishing servers designed to capture credentials.


3.0.5 Remediation

Change the password, keeping in mind proper password policy:

- Increase minimum password length to 12+ characters
- Require an uppercase character

- Require a special letter

By requiring longer passwords with uppercase and special characters, the keyspace is increased dramatically, making it much harder to guess using a simple dictionary attack.

 It is also very common for IoT devices or network edge devices to become compromised with malware. Therefore it is recommended to perform a firmware reset and upgrade the device in question to ensure its safe operation. For detailed instructions on a firmware reset or upgrade, visit the device manufacturer's website.

3.0.6 References

- 1: Wikipedia. Password strength
- 2: OWASP. Testing for Weak Password Policy

3.1 Unauthorized Access To Nextcloud

3.1.1 Affected Systems

- <https://nextcloud.example.com/>

3.1.2 Severity (CVSS)

CVSS 3.1 Vector: **10.0 - CRITICAL**

AC:L / **PR:N** / **UI:N** / **S:C** / **C:H** / **I:H** / **A:H**

3.1.3 Short Description

Due to a lax configuration of the Nextcloud instance, it was possible to access data without proper authorization. Sensitive information could be exposed and leaked by an attacker in this manner, and used for further attacks.

3.1.4 Technical Description

Nextcloud allows registration at <https://nextcloud.example.com/apps/registration/>.

Although only mail addresses from specific domains are allowed, an attacker can use an arbitrary address from the allowed domains. This is accepted by Nextcloud since the e-mail verification is not required.

After the registration, the files and folders shared by other users are readable by the attacker.

Among the discovered files were the following:

- OpenVPN configuration files
- Firmware builds for various products
- Virtual machine images
- Documents detailing internal development processes (bugfixes, roadmaps, etc.)
- Log files from numerous applications and servers
- Information detailing customer infrastructure
- Some recorded meetings
- Various packet capture files

Due to the sensitive nature of these files, some devastating attacks are possible from here:

- Use the OpenVPN files to gain entry into various internal networks.
- Gain knowlege of master password / key generation algorithms for some products by reading the python scripts.
- Develop malicious firmware for products by reversing the firmware images, or replacing the images with malware-laced firmware.
- Gain access to their macOS development machines by placing malicious payloads into shared folders.
- Attempt to use the information to infiltrate customer's networks

3.1.5 Remediation

Validate e-mail addresses used by Nextcloud and do not allow accessing any data without a verified e-mail address from the allowed domains. Alternatively, disable the registration altogether and create accounts on demand. Using fine-grained permissions to limit unnecessary exposure is also an important step towards reducing the associated risks. Furthermore, keeping your Nextcloud instance up to date is a good measure to avoid any publicly known exploits.

3.1.6 References

- 1: Nextcloud Server Hardening

3.2 Sensitive Data Exposure on SMB Shares

3.2.1 Affected Systems

- 192.168.1.150 (Fileshare Server)

3.2.2 Severity (CVSS)

CVSS 3.1 Vector: 6.4 – MEDIUM

AC:L / PR:N / UI:N / S:U / C:H / I:N / A:N

3.2.3 Short Description

The affected system allows an unauthenticated visitor to view sensitive files stored on the SMB shares.

3.2.4 Technical Description

When scanning the affected server for open ports, it was noticed that the port 445 for SMB was open. Upon further inspection, it became clear that the SMB service allowed an unauthenticated user to list the shares.

! Inside the fileshares it was possible to find sensitive information such as references to clients, financial documents and confidential information.

3.2.5 Remediation

Due to the low complexity and relatively big impact of the attack, our recommendation is to completely disallow unauthenticated or Guest users from interacting with the affected fileshares.

Furthermore, documents of sensitive or confidential nature should never be stored in easily accessible file shares, especially not without authentication. The reason is that the network perimeter can be breached by an attacker at any time, thus it is essential to secure the infrastructure in the internal network as well.

For hardening, you can also enable the Windows Group Policy “Do Not Allow Anonymous Enumeration of SAM Accounts and Shares” security setting.

3.2.6 References

- 1: MITRE ATT&CK T1135

3.3 Reflected XSS Vulnerability

3.3.1 Affected Systems

- webapp.dev.example.com

3.3.2 Severity (CVSS)

CVSS 3.1 Vector: 6.0 - MEDIUM

AC:L / PR:N / UI:R / S:C / C:L / I:L / A:N

3.3.3 Short Description

The affected web application is vulnerable to a Reflected Cross-Site-Scripting (XSS) attack. This can be used to send highly realistic phishing links to unsuspecting users.

3.3.4 Technical Description

The following link demonstrates the issue on the vulnerable page: [https://webapp.dev.example.com/app/images.html?imgurl=x"%20onerror=alert\(1\)>](https://webapp.dev.example.com/app/images.html?imgurl=x)

Visiting the link above will cause the visitor's browser to emit an alert box containing 1. This signifies the execution of the harmless JavaScript code for demonstration purposes.

An attacker can use this to create links/URLs similar to the one used to demonstrate the issue above. This can be used to create highly realistic phishing links that can steal the accounts of whoever visits the link. Due to the origin of the link being a trusted site, the users would be more likely to fall for the phishing attack.

The error in the code can be seen in the following snippet:

```
<?php
<html>
  <div>
    
  </div>
```

</html>

?>

The src attribute of the img tag is taking the imgurl parameter from the HTTP GET request sent to the server. Due to the missing sanitization of the untrusted input, it is possible to inject arbitrary JavaScript content.

3.3.5 Remediation

The software must always sanitize untrusted (e.g. user-provided) input values before reflecting them on the HTML. This can be achieved in multiple ways but it is recommended to use the language-specific functions to do this. For instance, in PHP the htmlspecialchars functions can be used for this purpose. This encodes all special characters before inserting them onto the page.

3.3.6 References

- 1: OWASP. Cross Site Scripting (XSS)
- 2: MITRE. CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

3.4 Kerberoastable Accounts in Active Directory

3.4.1 Affected Systems

- dc01.example.local
- dc02.example.local

3.4.2 Severity (CVSS)

CVSS 3.1 Vector: 4.0 - MEDIUM

AC:L / PR:L / UI:N / S:C / C:L / I:N / A:N

3.4.3 Short Description

The affected domains have multiple user accounts with an SPN value set. These accounts' password hashes can be subjected to an offline cracking attack.


3.4.4 Technical Description

This attack can be performed by an attacker with regular user privileges on the domain.

The affected domain has multiple user accounts with an SPN value set:

- `svc_mssql@example.local`
- `svc_webproxy@example.local`
- `svc_web@example.local`

Due to the presence of the SPN, it is possible to query the domain controllers' Kerberos Ticket Granting Service for a service ticket belonging to this account. The ticket is encrypted with the NTLM hash of this account's password. Due to that, it is possible to conduct offline hash cracking attacks against the service tickets.

 During the test it was possible to crack 2 out of the 3 passwords. This led to the takeover of two highly privileged service accounts which can be used to access various servers in the network.

3.4.5 Remediation


It should be considered if the presence of an SPN is required for the accounts to function within their context. If deemed unnecessary, the SPN can be removed. It is also recommended to change the password once this is done.

In cases where it is not practical to remove the SPN, the password security of the affected accounts must be increased significantly. Specifically, they should have a long and complicated passwords in order to limit the chances of a successful offline hash cracking attack. To further minimize the risk of compromise it's recommended to automatically and periodically rotate the the passwords.

Current NIST guidance includes the following:

- Favor length over complexity.
- Require user-generated passwords to be 8–64 characters long and auto-generated passwords to be 6–64 characters long.
- Permit the use of all ASCII characters, including spaces and emojis

Due to the high possibility of offline attacks, the passwords should be as long as possible. To stay on the safe side, it is recommended that the passwords be longer than 32 characters.

 For detection monitor for anomalous Kerberos activity, such as enabling Audit Kerberos Service Ticket Operations to log Kerberos TGS service ticket requests. Particularly investigate irregular patterns of activity (ex:

accounts making numerous requests, Event ID 4769, within a small time frame, especially if they also request RC4 encryption [Type 0x17]).

3.4.6 References

- 1: MITRE ATT&CK. Steal or Forge Kerberos Tickets: Kerberoasting
- 2: Active Directory Security. Detecting Kerberoasting Activity

3.5 Username Enumeration By Response Timing

3.5.1 Affected Systems

- webapp.dev.example.com

3.5.2 Severity (CVSS)

CVSS 3.1 Vector: 3.5 - LOW

AV:N / AC:H / PR:N / UI:N / S:U / C:L / I:N / A:N

3.5.3 Short Description

The affected application allows an attacker to enumerate valid usernames by measuring the response time of the webserver.

3.5.4 Technical Description

Sending the following HTTP request to the application causes the HTTP server to respond within 2000 milliseconds:

```
POST /api/login HTTP/1.1
Host: webapp.dev.example.com
Content-Type: application/json

{
  "username": "nonExistentUser",
  "password": "doesntMatter"
}
```

However when the username parameter matches a valid entry in the database, the HTTP server responds within 3000-4000 milliseconds:

```
POST /api/login HTTP/1.1
Host: webapp.dev.example.com
Content-Type: application/json

{
  "username": "admin",
  "password": "doesntMatter"
}
```

This delay has been shown to be reliably present when the username parameter is valid. Therefore the time delay can be used as an existence oracle for usernames.

In this manner it is possible for an attacker to test a list of usernames for validity, and e.g. use the results to launch a dictionary attack on valid accounts.

3.5.5 Remediation

In both cases whether a given username exists in the database or not, the application should take the same amount of time to respond.

3.5.6 References

- 1: CWE-204: Observable Response Discrepancy

3.6 Insecure Content Security Policy

3.6.1 Affected Systems

- webapp.dev.example.com

3.6.2 Severity (CVSS)

CVSS 3.1 Vector: 0.0 - LOW

AC:L / PR:N / UI:N / S:U / C:N / I:N / A:N

3.6.3 Short Description

The affected application does not utilize a Content Security Policy (CSP).

3.6.4 Technical Description

A Content Security Policy (CSP) is a security feature that helps prevent a variety of attacks, including Cross-Site Scripting (XSS) and data injection attacks. By not implementing a CSP, the application is vulnerable to malicious scripts being executed in the context of the user's browser, which can lead to data theft, session hijacking, and other security issues.

The absence of a CSP means that the application does not restrict the sources from which content can be loaded. This can allow attackers to inject malicious scripts or load resources from untrusted sources, compromising the integrity and confidentiality of the application.

3.6.5 Remediation

To mitigate the risks associated with the lack of a Content Security Policy, the following steps should be taken:

1. **Implement a Content Security Policy:** Define a CSP that specifies which sources of content are trusted. This can include directives for scripts, styles, images, and other resources. A basic example of a CSP header might look like this:

```
Content-Security-Policy: default-src 'self'; script-src 'self' https://trusted.cdn.com; style-src 'self' 'unsafe-inline';
```

1. **Test the Policy:** Use tools like CSP Evaluator or browser developer tools to test the effectiveness of the CSP. Ensure that legitimate content is still loading while blocking potentially harmful content.
2. **Monitor and Update:** Regularly review and update the CSP as the application evolves. Ensure that any new features or third-party integrations are accounted for in the policy.
3. **Consider Reporting:** Implement a reporting mechanism to monitor CSP violations. This can help identify potential attacks or misconfigurations.

3.6.6 References

- 1: OWASP XSS Prevention Cheat Sheet
- 2: Content Security Policy (CSP) - MDN Web Docs
- 3: CSP Evaluator - Google
- 4: CSP Reporting - MDN Web Docs

4 Appendix

4.1 CVSS Explanation

The Common Vulnerability Scoring System (CVSS) is a standardized framework for rating the severity of security vulnerabilities in software. A CVSS vector is a

string that encodes the various metrics used to calculate the CVSS score. This score helps organizations prioritize their responses to vulnerabilities based on their severity.

4.1.1 Components of a CVSS Vector

A CVSS vector consists of several metrics, which can be categorized into three groups: Base, Temporal, and Environmental.

4.1.2 1. Base Metrics

The Base metrics represent the intrinsic characteristics of a vulnerability that are constant over time and across user environments. The Base score is the primary score and is calculated from the following metrics:

- **Attack Vector (AV):** The context by which an attacker can exploit the vulnerability.
 - N: Network
 - A: Adjacent
 - L: Local
 - P: Physical
- **Attack Complexity (AC):** The conditions beyond the attacker's control that must exist in order to exploit the vulnerability.
 - L: Low
 - H: High
- **Privileges Required (PR):** The level of privileges an attacker must possess before successfully exploiting the vulnerability.
 - N: None
 - L: Low
 - H: High
- **User Interaction (UI):** Whether the exploitation of the vulnerability requires any user interaction.
 - N: None
 - R: Required
- **Scope (S):** The extent of the impact of the vulnerability.
 - U: Unchanged
 - C: Changed
- **Confidentiality (C):** The impact on the confidentiality of the information.
 - N: None
 - L: Low
 - H: High
- **Integrity (I):** The impact on the integrity of the information.
 - N: None
 - L: Low
 - H: High
- **Availability (A):** The impact on the availability of the system.
 - N: None
 - L: Low
 - H: High

4.1.2.1 2. Temporal Metrics

The Temporal metrics measure the characteristics of a vulnerability that may change over time. These metrics include:

- **Exploit Code Maturity (E):** The current state of exploit techniques.
 - X: Not defined
 - U: Unproven
 - P: Proof-of-Concept
 - F: Functional
 - H: High
- **Remediation Level (RL):** The level of remediation available for the vulnerability.
 - X: Not defined
 - O: Official Fix
 - T: Temporary Fix
 - W: Workaround
 - U: Unavailable
- **Report Confidence (RC):** The degree of confidence in the existence of the vulnerability and the credibility of the report.
 - X: Not defined
 - U: Unknown
 - R: Reasonable
 - C: Confirmed

4.1.2.2 3. Environmental Metrics

The Environmental metrics allow organizations to customize the CVSS score based on their specific environment. These metrics include:

- **Modified Attack Vector (MAV)**
- **Modified Attack Complexity (MAC)**
- **Modified Privileges Required (MPR)**
- **Modified User Interaction (MUI)**
- **Modified Scope (MS)**
- **Modified Confidentiality (MC)**
- **Modified Integrity (MI)**
- **Modified Availability (MA)**

4.1.2.3 Example CVSS Vector

A CVSS vector might look like this:

AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

This vector indicates:

- **Attack Vector:** Network
- **Attack Complexity:** Low
- **Privileges Required:** None

- **User Interaction:** None
- **Scope:** Unchanged
- **Confidentiality Impact:** High
- **Integrity Impact:** High
- **Availability Impact:** High

Understanding CVSS vectors is crucial for assessing the severity of vulnerabilities and prioritizing remediation efforts. By analyzing the Base, Temporal, and Environmental metrics, organizations can make informed decisions about their security posture.

5 Password Security Glossary

- **Entropy:** how unpredictable or “random” your password is. You can calculate it using

$$H = L \times \log_2(C)$$

where L is your password’s length and C is the number of possible symbols (e.g., 94 for standard keyboards). Each bit of entropy doubles the number of guesses a hacker needs. For example, a 10-character password using 94 symbols has 65.5 bits of entropy. At 10^9 guesses per second, this would take 3,800 years to crack—if the password is truly random. But if you use a predictable pattern like “P@ssw0rd,” entropy becomes irrelevant. Hackers recognize these patterns, cracking them in seconds. Entropy only works if your password isn’t predictable.

- **Brute-Force Attacks:** tries every possible password combination. The average time to crack a password is

$$T = C^L / (2 \times G)$$

, where C^L is the total combinations, and G is guesses per second. For a 10-character password with 94 symbols, that’s 94^{10} combinations—about 3,800 years for a hacker with a fast GPU. But hackers rarely brute-force blindly. They exploit shortcuts, like prioritizing common patterns (e.g., “12345”), allowing them to skip 99% of possibilities. The math assumes worst-case security, but real-world attackers are far more efficient.

- **Dictionary Attacks:** combines words from a predefined list (e.g., “cat”) with numbers or symbols (e.g., “2023”). The math is simple:

$$\text{Combos} = D \times S$$

, where D is the dictionary size (10^5 words) and S is possible suffixes (10^4 numbers/symbols). A password like “cat2023” has 10^9 combinations, which a hacker can crack in 1 second at 10^9 guesses/second. Even if you think adding a year makes it unique, it’s still vulnerable. This attack thrives on the human tendency to reuse familiar phrases.

- **Rule-Based Attacks:** use predictable substitutions, like replacing “E” with “3” or adding “123” to the end. There’s no complex math here, hackers pre-program these rules into their tools. For example, “P@ssw0rd123” might seem clever, but it’s a textbook example in hacker databases. These rules drastically reduce the number of guesses needed, turning a “strong” password into an easy target in seconds.
- **Hash Rate:** G is the number of passwords a hacker can test per second. Assume single consumer GPU can process 10^9 guesses/second. With a cluster of 100 GPUs, that jumps to 10^{11} guesses/second. At this rate, an 8-character password (94 symbols) with 52 bits of entropy would take 14 hours to crack. But if the password uses fewer symbols (e.g., only lowercase letters), it falls in minutes. This is why length and randomness matter—each extra character multiplies the hacker’s workload.
- **Unicode:** using symbols like 🍌, ö, or 日本, expand the pool of possible characters C from 94 to 155063. The entropy formula $H = L \times \log_2(C)$ becomes far more powerful. An 8-character Unicode password has 135 bits of entropy—equivalent to a 22-character ASCII password. Since most hackers assume standard keyboards, they’d need to brute-force $(1.5 \times 10^5)^8$ combinations. At 10^{12} guesses/second, this would take billions of years. Unicode’s diversity makes it a formidable defense, but only if attackers haven’t adapted their tools.

5.0.1 Why This Matters

A “strong” password is about being random *and* long enough. Hackers exploit laziness, not math.

5.0.2 How to Protect Yourself

1. Use **12+ random characters** (mix letters, numbers, symbols) or **8+ Unicode symbols**.
2. Let a **password manager** generate and store passwords.
3. **Never reuse passwords** else a single breach compromise multiple accounts.