

PROJECT REPORT

CYBER SECURITY

NETWORK VULNERABILITY ASSESSMENT



TEAM MEMBERS

TEAM LEADER: MAYANK RAJ

REG NO: 720134205204

TEAM MEMBER: KUCHARLAPATI
VAMSI KRISHNA

REG NO: 720134205203

TEAM MEMBER: KODURU
VENKATESHWARLU

REG NO: 720134205202

TEAM MEMBER: GORLA RAHUL
REG NO: 720134205198

INTRODUCTION

A network vulnerability assessment reviews and analyzes an organization's network infrastructure to find cybersecurity vulnerabilities and network security loopholes. The evaluation can be done manually or by using vulnerability analysis software. Software is preferred because it's less susceptible to human error and delivers more accurate results.

A vulnerability assessment determines the strength of a company's network security. Along the way, it uncovers any security vulnerabilities that might compromise the overall business operations, cybersecurity, compliance, and network privacy.

Vulnerability assessments should not be confused with penetration tests (also known as pen tests), which simulate a cyber attack. A penetration test does typically focus on firewalls and their vulnerabilities, since those are the best ways to "crack" a secure network but the simulated attacks of a pen test and the analysis of a vulnerability assessment are not the same thing.

IMPORTANCE

The most common security vulnerabilities are rooted either in technology issues or user behavior:

1. Breaches can occur if insiders accidentally expose information to an external source or leak information intentionally (i.e., malicious insiders).
 2. Lost and stolen devices that contain unencrypted data are also a major vector for infiltration into a company's network.
 3. Cybercriminals can install malware on target systems to exfiltrate data or gain control over computing systems.
- The process is ongoing and involves conducting periodic vulnerability assessments - when one assessment completes, another must begin.

Vulnerability assessments allow security teams to identify, analyze, categorize, report, and remediate security vulnerabilities in operating systems, business applications, endpoint devices, and browsers.

OVERVIEW

Vulnerability scanning is the process of discovering, analyzing, and reporting on security flaws and vulnerabilities.

Vulnerability scans are conducted via automated vulnerability scanning tools to identify potential risk exposures and attack vectors across an organization's networks, hardware, software, and systems.



Common network vulnerabilities

1. Shadow IT

Shadow IT refers to any systems, applications, or services that were installed without prior authorization from the designated representative of the organization's IT department. These systems have not thus been monitored and managed properly, nor have they undergone any form of security testing.

2. Mobile device

Mobile devices are increasingly being used to access corporate networks, namely through Bring Your Own Device (BYOD) policies. This can be a security issue because mobile devices may not have the same level of security features as computers, such as an antivirus or a firewall.

3. Poor firewall configurations

Gartner Research states that misconfigurations are causing 95% of all firewall breaches. A firewall is a network security system that controls the incoming and outgoing traffic of a network. It is used to block off or allow certain types of traffic based on predefined rules.

4. Insecure incoming emails

One of the most common network security issues is email-based attacks. These attacks can take many forms, but the most common is phishing. Phishing is a type of email attack that uses fraudulent emails to trick users into revealing

sensitive information such as passwords or credit card numbers.

5. Weak passwords

If you're using a weak or default password, chances are your network is already compromised. A strong password is at least 8 characters long and includes a mix of uppercase and lowercase letters, numbers, and symbols.

6. Insecure wireless networks

When configured by default or with a weaker, obsolete security protocol such as WEP or WPA2 PSK (pre-shared key), any wireless access points will appear to outside attackers as an open invitation to launch malware, data theft, or security breach attacks.

1) INFORMATION GATHERING:

Information gathering is a crucial phase in the cybersecurity and assessment process. It involves collecting relevant data and intelligence about a target system, network, or organization to understand its vulnerabilities and potential attack surfaces. Here are different aspects of information gathering:

a) EMAIL FOOTPRINT ANALYSIS:

Email footprint refers to collecting information from emails by monitoring the email delivery and inspecting the headers.

Information collected through email footprint includes:

- IP address of the recipient
- Geolocation of the recipient
- Delivery information
- Visited links
- Browser and OS information
- Reading time

Email footprint analysis involves collecting information related to an organization's email infrastructure, such as email addresses, email servers, and email security measures. This analysis helps in understanding how email communications are handled and identifying potential points of entry for attackers.

— Domain Profile

Registrar	CSC CORPORATE DOMAINS, INC. CSC Corporate Domains, Inc. IANA ID: 299 URL: www.cscprotectsbrands.com , http://cscdbs.com Whois Server: whois.corporatedomains.com domainabuse@cscglobal.com (p) +1.8887802723
Registrar Status	clientTransferProhibited

Domain Name: testfire.net
Registry Domain ID: 8363973_DOMAIN_NET-VRSN
Registrar WHOIS Server: whois.corporatedomains.com
Registrar URL: www.cscprotectsbrands.com
http://cscdbs.com
Updated Date: 2023-07-19T01:05:02+00:00
2023-07-19
Creation Date: 1999-07-23T09:52:32+00:00
1999-07-23
Registrar Registration Expiration Date: 2024-07-
23T13:52:32+00:00
2024-07-23
Registrar: CSC CORPORATE DOMAINS, INC.
CSC Corporate Domains, Inc.
Sponsoring Registrar IANA ID: 299
Registrar Abuse Contact Email: domainabuse@cscglobal.com
Registrar Abuse Contact Phone: +1.8887802723
Status:

b) DNS GATHERING INFORMATION:

DNS (Domain Name System) information gathering involves querying and analyzing DNS records to gather details about domain names, IP addresses, mail exchange servers, and other crucial information. It helps in understanding the network structure and identifying potential targets for cyberattacks.

When we talk about DNS enumeration, we generally refer to every technique we use to gather as much data as possible by querying the DNS server of a website or host. DNS is regularly a rewarding hotspot for active data gathering. M

The screenshot shows a web page titled "Domain Name System (DNS) Historical Record Archive". The main content area displays historical DNS records for the domain `testfire.net`. The records are categorized into sections: DNS Records, SOA, NS, PNS, MX, A, AAAA, CNAME, PTR, and TXT. The DNS Records section shows the domain was added in 2009 and last updated in 2023. The SOA section provides details like Refresh, Retry, and Expire times. The NS section lists multiple nameservers over time. The PNS section shows previous nameservers. The MX section lists one MX record. The A section lists one IP address. The AAAA section lists one IPv6 address. The CNAME section lists one alias. The PTR section lists one reverse DNS entry. The TXT section lists one SPF record.

DNS History

Domain Name System (DNS) Historical Record Archive.

With our new look website you can now find other domains hosted on the same IP address, your website neighbours and more even quicker than before.

prev next Domain Search Search [Donate](#)

DNS Records

Domain: `testfire.net`.
Added: 2009-07-21
Last updated: 2023-07-30

What points here by: [CNAME](#) / [NS](#) / [MX](#) / [PTR](#)
[View SubDomains](#) / [Check DNS Propagation](#) / [Dig](#).

SOA - (History:5)

2022-06-24 -> 2023-07-30
MName: `asia3.akam.net`
Serial: 1366025607
Refresh: 43200
Retry: 7200
Expire: 86400

NS - (History:34)

2022-06-24 -> 2023-07-30 `eur5.akam.net`
2022-06-24 -> 2023-07-30 `nsl-99.akam.net`
2022-06-24 -> 2023-07-30 `eur2.akam.net`
2022-06-24 -> 2023-07-30 `usc2.akam.net`
2022-06-24 -> 2023-07-30 `nsl-206.akam.net`
2022-06-24 -> 2023-07-30 `usw2.akam.net`
2022-06-24 -> 2023-07-30 `usc3.akam.net`
2022-06-24 -> 2023-07-30 `asia3.akam.net`

PNS

2021-06-04 -> 2021-06-04 `asia3.akam.net`
2021-06-04 -> 2021-06-04 `eur2.akam.net`
2021-06-04 -> 2021-06-04 `eur5.akam.net`
2021-06-04 -> 2021-06-04 `nsl-206.akam.net`
2021-06-04 -> 2021-06-04 `nsl-99.akam.net`
2021-06-04 -> 2021-06-04 `usc2.akam.net`
2021-06-04 -> 2021-06-04 `usc3.akam.net`
2021-06-04 -> 2021-06-04 `usw2.akam.net`

MX

A - (History:3)

2022-06-24 -> 2023-07-30 `65.61.137.117`

AAAA

CNAME

PTR

TXT - (History:1)

2022-06-24 -> 2023-07-30 `v=spf1 mx/24 -all`

c) **WHOIS INFORMATION GATHERING**

WHOIS information gathering involves querying the WHOIS database to retrieve registration details of domain names and IP addresses. This data includes contact information of domain owners and registrars, which can be valuable for understanding the ownership and potential affiliations of a target domain.

Whois is the name of the protocol that is used to interrogate the servers operated by Regional Internet Registries, which hold information about every resource (IP address or domain name) registered on the Internet.

The information that you can obtain about a resource is:

- Name of the owner company
- Address of the owner company
- The IP range that a certain IP belongs to
- Contact phone number
- Contact email
- Administrator's name
- Name servers

Whois Record (last updated on 2023-07-29)

Domain Name: testfire.net
Registry Domain ID: 8363973_DOMAIN_NET-VRSN
Registrar WHOIS Server: whois.corporatedomains.com
Registrar URL: www.cscprotectsbrands.com
http://cscdbs.com
Updated Date: 2023-07-19T01:05:02+00:00
2023-07-19
Creation Date: 1999-07-23T09:52:32+00:00
1999-07-23
Registrar Registration Expiration Date: 2024-07-
23T13:52:32+00:00
2024-07-23
Registrar: CSC CORPORATE DOMAINS, INC.
CSC Corporate Domains, Inc.
Sponsoring Registrar IANA ID: 299
Registrar Abuse Contact Email: domainabuse@cscglobal.com
Registrar Abuse Contact Phone: +1.8887802723
Status:
clientTransferProhibited
Registry Registrant ID:
Registrant Name: REDACTED FOR PRIVACY (DT)
Registrant Organization: Not Disclosed
Registrant Street: Not Disclosed
Registrant City: Sunnyvale
Registrant State/Province: CA
Registrant Postal Code: 94085
Registrant Country: US
Registrant Phone: +Not Disclosed
Registrant Phone Ext:
Registrant Fax: +Not Disclosed
Registrant Fax Ext:
Registrant Email: REDACTED FOR PRIVACY (DT)
Registry Admin ID:
Admin Name: REDACTED FOR PRIVACY (DT)
Admin Organization: Not Disclosed
Admin Street: Not Disclosed
Admin City: Sunnyvale
Admin State/Province: CA
Admin Postal Code: 94085
Admin Country: US
Admin Phone: REDACTED FOR PRIVACY (DT)
Admin Phone Ext:
Admin Fax: REDACTED FOR PRIVACY (DT)
Admin Fax Ext:
Admin Email: REDACTED FOR PRIVACY (DT)
Registry Tech ID:
Tech Name: REDACTED FOR PRIVACY (DT)
Tech Organization: Not Disclosed
Tech Street: Not Disclosed
Tech City: Sunnyvale
Tech State/Province: CA
Tech Postal Code: 94085
Tech Country: US
Tech Phone: REDACTED FOR PRIVACY (DT)

Tech Phone Ext:
Tech Fax: REDACTED FOR PRIVACY (DT)
Tech Fax Ext:
Tech Email: REDACTED FOR PRIVACY (DT)
Registry Billing ID:
Billing Name:
Billing Organization:
Billing Street:
Billing City:
Billing State/Province:
Billing Postal Code:
Billing Country:
Billing Phone:
Billing Phone Ext:
Billing Fax:
Billing Fax Ext:
Billing Email:
Nameservers:
 asia3.akam.net
 eur2.akam.net
 eur5.akam.net
 ns1-206.akam.net
 ns1-99.akam.net
 usc2.akam.net
 usc3.akam.net
 usw2.akam.net
DNSSEC: unsigned

d) INFORMATION GATHERING FOR SOCIAL ENGINEERING ATTACKS

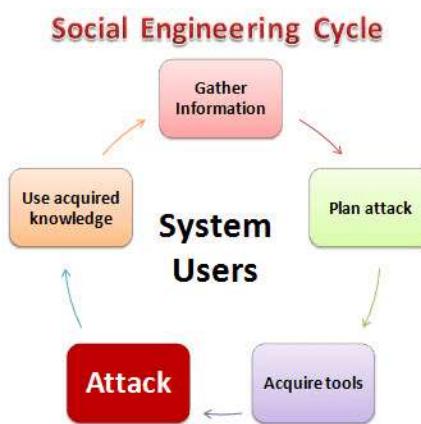
WHOIS information gathering involves querying the WHOIS database to retrieve

registration details of domain names and IP addresses. This data includes contact information of domain owners and registrars, which can be valuable for understanding the

ownership and potential affiliations of a target domain.

Social engineering is the tactic of manipulating, influencing, or deceiving a victim in order to gain control over a computer system, or to steal personal and financial information. It uses psychological manipulation to trick users into making security mistakes or giving away sensitive information.

Social engineering attacks happen in one or more steps. A perpetrator first investigates the intended victim to gather necessary background information, such as potential points of entry and weak security protocols, needed to proceed with the attack.



e) EMERGING TRENDS AND TECHNOLOGIES IN INFORMATION GATHERING:

As technology evolves, so do the methods of information gathering. Emerging trends include the use of artificial intelligence and machine learning algorithms for automated data collection and analysis, advanced OSINT (Open-Source Intelligence) tools, and social media analysis for gathering valuable intelligence.

The result of the information gathering performed on
domain name : testfire.net

2) VULNERABILITY IDENTIFICATION:

The assessment follows a well-defined methodology, including the following steps

1. Reconnaissance: Passive reconnaissance techniques are used to gather information about the network and its assets.
2. Vulnerability Scanning: Automated scanning tools are employed to identify potential vulnerabilities in the network.

3. Manual Verification: The identified vulnerabilities are manually verified to eliminate false positives and prioritize critical issues.
4. Exploitation (with Authorization): Ethical exploitation of vulnerabilities is conducted to determine the extent of potential damage if exploited maliciously.
5. Analysis and Reporting: The assessment findings are analyzed, and a detailed report is generated, including a list of vulnerabilities, risk severity, and actionable recommendations.
Analysts also rely on vulnerability databases, vendor vulnerability announcements, asset management systems and threat intelligence feeds to identify security weaknesses.



a) IDENTIFY AND NAME EACH VULNERABILITY:

A vulnerability is a hole or a weakness in the application, which can be a design flaw or an implementation bug, that allows an attacker to cause harm to the stakeholders of an application

To identify the vulnerability of any website.

There were many tools like Nikto, SQL map, burp suite, etc....

TARGET: <http://testfire.net/>

The following website is tested by nikto.

\$nikto -h <http://testfire.net/>

1. Identify Vulnerabilities: The assessment aims to identify potential vulnerabilities in the network infrastructure, including unpatched software, misconfigurations, and open ports.
2. Evaluate Security Controls: The effectiveness of existing security controls, such as firewalls, intrusion detection systems (IDS), and access controls, is assessed to determine their ability to detect and prevent attacks.
3. Assess Network Architecture: The network architecture is reviewed to ensure proper segmentation, isolation of critical assets, and a robust perimeter defense.
4. Password Policy Evaluation: The assessment examines the strength of password

policies and their adherence to industry best practices to prevent unauthorized access.

5. Physical Security Analysis: Physical security measures in place to protect network infrastructure and data centers are evaluated to prevent unauthorized physical access.

```
root@localhost:/home/userland# nikto -h http://testfire.net/
- Nikto v2.1.5
-----
+ Target IP:      65.61.137.117
+ Target Hostname: testfire.net
+ Target Port:    80
+ Start Time:    2023-08-08 17:08:56 (GMT0)
-----
+ Server: Apache-Coyote/1.1
+ Cookie JSESSIONID created without the httponly flag
+ The anti-clickjacking X-Frame-Options header is not present.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Allowed HTTP Methods: GET, HEAD, POST, PUT, DELETE, OPTIONS
+ OSVDB-397: HTTP method ('Allow' Header): 'PUT' method could allow clients to save files on the web server.
+ OSVDB-5646: HTTP method ('Allow' Header): 'DELETE' may allow clients to remove files on the web server.
+ DEBUG HTTP verb may show server debugging information. See http://msdn.microsoft.com/en-us/library/e8z01xdh%28VS.80%29.aspx for details.
```

b) ASSIGN A COMMON WEAKNESS ENUMERATION(CWE) CODE TO EACH VULNERABILITY:

Targeted at both the development community and the community of security practitioners, Common Weakness Enumeration (CWE™) is a formal list or dictionary of common software and hardware weaknesses that can occur in architecture, design, code, or implementation that can lead to exploitable security vulnerabilities. CWE was created to serve as a common language for describing security weaknesses; serve as a standard measuring stick for security tools targeting these weaknesses; and provide a common baseline standard for weakness identification, mitigation, and prevention efforts.

- nmap testphp.vulnweb.com
- Starting Nmap 7.94 (<https://nmap.org>) at 2023-07-14 13:50 IST
- Nmap scan report for testphp.vulnweb.com
- (44.228.249.3)
- Host is up (0.28s latency).
- CVSSV3.1:
- rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
- Not shown: 999 filtered tcp ports (no-response) CVE-ID: CVE-2022-417
- PORT STATE SERVICE
- CWE-ID: CWE-125-Ouch

- 80/tcp open http
- Exploit availability: No
- Nmap done: 1 IP address (1 host up) scanned in 21.04 seconds
- Step -3
- Take ip address of domain and
- nmap -sV 44.228.249.3 -p 80
- nginx 1.0.7-1.23.1
- \$ nmap -sV 44.228.249.3 -p 80
- Starting Nmap 7.94 (<https://nmap.org>) at 2023-07-14 13:51 IST
- ec2-44-228-249-3.us-west-2.compute.amazonaws.com (44.228.249.3)
- Nmap scan report for
- Host is up (0.30s latency).
- PORT STATE SERVICE VERSION
- 80/tcp open http nginx 1.19.0
- che 2.3 an
- External links
- Service detection performed. Please report any incorrect results at <https://nmap.org/submit/>
- Nmap done: 1 IP address (1 host up) scanned in 23.10 seconds

- Step -4
- Copied the version
- nginx 1.19.0 and
- Pasted it in google

- The results is
- # PHuiP-FPizdaM
- ## What's this
- This is an exploit for a bug in php-fpm (CVE-2019-11043). In certain nginx + php-fpm configurations, the bug is possible to trigger from the outside. This means that a web user may get code execution if you have vulnerable config (see [below](#the-full-list-of-preconditions)).
- ## What's vulnerable
- If a webserver runs nginx + php-fpm and nginx have a configuration like
- `location ~ [^/]/\.php(/|\$) {`
- `...`
- `fastcgi_split_path_info ^(.+?\.\php)(/.*)$;`
- `fastcgi_param PATH_INFO`
- `$fastcgi_path_info;`
- `fastcgi_pass php:9000;`
- `...`
- }which also lacks any script existence checks (like `try_files`), then you can probably hack it with this exploit.
- ##### The full list of preconditions
- 1. Nginx + php-fpm, `location ~ [^/]\.\php(/|\$)` must be forwarded to php-fpm (maybe the regexp can be stricter, see [#1](<https://github.com/neex/phuip-fpizdam/issues/1>)).

- 2. The fastcgi_split_path_info directive must be there and contain a regexp starting with ^ and ending with \$, so we can break it with a newline character.
- 3. There must be a PATH_INFO variable assignment via statement fastcgi_param PATH_INFO \$fastcgi_path_info;. At first, we thought it is always present in the fastcgi_params file, but it's not true.
- 4. No file existence checks like try_files \$uri =404 or if (-f \$uri). If Nginx drops requests to non-existing scripts before FastCGI forwarding, our requests never reach php-fpm. Adding this is also the easiest way to patch.
- 5. This exploit works only for PHP 7+, but the bug itself is present in earlier versions (see [below](#about-php5)).
- ## Isn't this known to be vulnerable for years?
- A long time ago php-fpm didn't restrict the extensions of the scripts, meaning that something like /avatar.png/some-fake-shit.php could execute avatar.png as a PHP script. This issue was fixed around 2010.
-
- The current one doesn't require file upload, works in the most recent versions (until the fix has landed), and, most importantly, the exploit is much cooler.
- ## How to run

- Install it using
- go get github.com/neex/phuip-fpizdam
- If you get strange compilation errors, make sure you're using go >= 1.13. Run the program using phuip-fpizdam [url] (assuming you have the \$GOPATH/bin inside your \$PATH, otherwise specify the full path to the binary).
Good output looks like this:
- 2019/10/01 02:46:15 Base status code is 200
- 2019/10/01 02:46:15 Status code 500 for qsl=1745, adding as a candidate
- 2019/10/01 02:46:15 The target is probably vulnerable. Possible QSLs: [1735 1740 1745]
- 2019/10/01 02:46:16 Attack params found: --qsl 1735 --pisos 126 --skip-detect
- 2019/10/01 02:46:16 Trying to set "session.auto_start=0"...
- 2019/10/01 02:46:16 Detect() returned attack params: --qsl 1735 --pisos 126 --skip-detect <-- REMEMBER THIS
- 2019/10/01 02:46:16 Performing attack using php.ini settings...
- 2019/10/01 02:46:40 Success! Was able to execute a command by appending "?a=/bin/sh+-c+'which+which'&" to URLs
- 2019/10/01 02:46:40 Trying to cleanup /tmp/a...
- 2019/10/01 02:46:40 Done!

- `After this, you can start appending `?a=<your command>` to all PHP scripts (you may need multiple retries).
- ## Playground environment
- If you want to reproduce the issue or play with the exploit locally, do the following:
 - 1. Clone this repo and go to the `reproducer` directory.
 - 2. Create the docker image using `docker build -t reproduce-cve-2019-11043 .`. It takes a long time as it internally clones the php repository and builds it from the source. However, it will be easier this way if you want to debug the exploit. The revision built is the one right before the fix.
 - 2. Run the docker using `docker run --rm -ti -p 8080:80 reproduce-cve-2019-11043`.
 - 3. Now you have <http://127.0.0.1:8080/script.php>, which is an empty file.
 - 4. Run the exploit using `phuip-fpizdam <http://127.0.0.1:8080/script.php>`
 - 5. If everything is ok, you'll be able to execute commands by appending `?a=` to the script: <http://127.0.0.1:8080/script.php?a=id>. Try multiple times as only some of php-fpm workers are infected.
- ## About PHP5

- The buffer underflow in php-fpm is present in PHP version 5. However, this exploit makes use of an optimization used for storing FastCGI variables, [\[_fcgi_data_seg\]](https://github.com/php/php-src/blob/5d6e923/main/fastcgi.c#L186). This optimization is present only in php 7, so this particular exploit works only for php 7. There might be another exploitation technique that works in php 5.

- ## Credits

Original anomaly discovered by [d90pwn] (<https://twitter.com/d90pwn>) during Real World CTF. Root clause found by me (Emil Lerner) as well as the way to set php.inioptions. Final php.ini options set is found by [beched] (https://twitter.com/ahack_ru).

Identifying and naming vulnerabilities involves the process of discovering and documenting specific security weaknesses or flaws in an application. This process typically involves using automated tools or manual testing techniques to identify potential vulnerabilities. Once a vulnerability has been identified, it must be given a descriptive name that accurately reflects the nature of the vulnerability.

c) **PROVIDE CORRESPONDING Open Web Application Security Project (OWASP) Category And Description For Each Vulnerability:**

The Open Web Application Security Project (OWASP) is a nonprofit foundation that provides guidance on how to develop, purchase and maintain trustworthy and secure software applications. OWASP is noted for its popular Assigning CWE codes to each vulnerability is an essential step in the vulnerability identification process. A CWE code is a unique identifier assigned to a specific type of vulnerability, making it easier to identify and categorize similar types of vulnerabilities. Assigning a CWE code to each vulnerability allows developers and security professionals to more easily track, analyze and remediate potential security issues.

- [**A01:2021-Broken Access Control**](#) moves up from the fifth position to the category with the most serious web application security risk; the contributed data indicates that on average, 3.81% of applications tested had one or more Common Weakness Enumerations (CWEs) with more than 318k occurrences of CWEs in this risk category. The 34 CWEs mapped to Broken Access Control had more occurrences in applications than any other category.

- **A02:2021-Cryptographic Failures** shifts up one position to #2, previously known as **A3:2017-Sensitive Data Exposure**, which was broad symptom rather than a root cause. The renewed name focuses on failures related to cryptography as it has been implicitly before. This category often leads to sensitive data exposure or system compromise.
- **A03:2021-Injection** slides down to the third position. 94% of the applications were tested for some form of injection with a max incidence rate of 19%, an average incidence rate of 3.37%, and the 33 CWEs mapped into this category have the second most occurrences in applications with 274k occurrences. Cross-site Scripting is now part of this category in this edition.
- **A04:2021-Insecure Design** is a new category for 2021, with a focus on risks related to design flaws. If we genuinely want to "move left" as an industry, we need more threat modeling, secure design patterns and principles, and reference architectures. An insecure design cannot be fixed by a perfect implementation as by definition, needed security controls were never created to defend against specific attacks.
- **A05:2021-Security Misconfiguration** moves up from #6 in the previous edition; 90% of applications were tested for some form of misconfiguration, with an average incidence rate of 4.5%, and over 208k occurrences of CWEs mapped to this risk category. With more shifts into highly configurable software,

it's not surprising to see this category move up. The former category for **A4:2017-XML External Entities (XXE)** is now part of this risk category.

- **A06:2021-Vulnerable and Outdated Components** was previously titled Using Components with Known Vulnerabilities and is #2 in the Top 10 community survey, but also had enough data to make the Top 10 via data analysis. This category moves up from #9 in 2017 and is a known issue that we struggle to test and assess risk. It is the only category not to have any Common Vulnerability and Exposures (CVEs) mapped to the included CWEs, so a default exploit and impact weights of 5.0 are factored into their scores.
- **A07:2021-Identification and Authentication Failures** was previously Broken Authentication and is sliding down from the second position, and now includes CWEs that are more related to identification failures. This category is still an integral part of the Top 10, but the increased availability of standardized frameworks seems to be helping.
- **A08:2021-Software and Data Integrity Failures** is a new category for 2021, focusing on making assumptions related to software updates, critical data, and CI/CD pipelines without verifying integrity. One of the highest weighted impacts from Common Vulnerability and Exposures/Common Vulnerability Scoring System (CVE/CVSS) data mapped to the 10 CWEs in this category. **A8:2017-Insecure Deserialization** is now a part of this larger category.

- **A09:2021-Security Logging and Monitoring Failures** was previously **A10:2017-Insufficient Logging & Monitoring** and is added from the Top 10 community survey (#3), moving up from #10 previously. This category is expanded to include more types of failures, is challenging to test for, and isn't well represented in the CVE/CVSS data. However, failures in this category can directly impact visibility, incident alerting, and forensics.
- **A10:2021-Server-Side Request Forgery** is added from the Top 10 community survey (#1). The data shows a relatively low incidence rate with above average testing coverage, along with above-average ratings for Exploit and Impact potential. This category represents the scenario where the security community members are telling us this is important, even though it's not illustrated in the data at this time.



d) UNDERSTANDING AND DEFINING VULNERABILITIES:

Understanding and defining vulnerabilities is a critical first step in identifying and mitigating potential risks in an application. A vulnerability can be defined as a flaw or weakness in the system that can be exploited by attackers to compromise the security of the system.

Vulnerabilities can exist in different layers of the application, including the network layer, application layer, and database layer. By understanding the different types of vulnerabilities that exist, developers and security professionals can take appropriate measures to mitigate the risks and prevent attacks.

A vulnerability is a weakness in an IT system that can be exploited by an attacker to deliver a successful attack. They can occur through flaws, features, or user error, and attackers will look to exploit any of them, often combining one or more, to achieve their end goal.

A feature is intended functionality which can be misused by an attacker to breach a system. Features may improve the user's experience, help diagnose problems or improve management, but they can also be exploited by an attacker.

When Microsoft introduced macros into their Office suite in the late 1990s, macros soon became the vulnerability of choice with the Melissa worm in 1999 being a prime example. Macros are still exploited today; the Dridex banking Trojan that was spreading in late 2014 relies on spam to

deliver Microsoft Word documents containing malicious macro code, which then downloads Dridex onto the affected system.

JavaScript, widely used in dynamic web content, continues to be used by attackers. This includes diverting the user's browser to a malicious website and silently downloading malware, and hiding malicious code to pass through basic web filtering.

e) IDENTIFYING AND NAMING THE VULNERABILITIES:

Identifying and naming vulnerabilities is the next step in the vulnerability assessment process. This involves conducting a thorough analysis of the application to identify all potential vulnerabilities that could be exploited by attackers. Once identified, each vulnerability should be given a clear and concise name that accurately describes the nature of the vulnerability.

A vulnerability naming scheme is a systematic method for creating and maintaining a standardized dictionary of common names for a set of vulnerabilities in IT systems, such as software flaws in an operating system or security configuration issues in an application. The naming scheme ensures that each vulnerability entered into the dictionary has a unique name. Using standardized vulnerability naming

schemes supports interoperability. Organizations typically have many tools for system security management that reference vulnerabilities—for example, vulnerability and patch management software, vulnerability assessment tools, antivirus software, and intrusion detection systems. If these tools do not use standardized names for vulnerabilities, it may not be clear that multiple tools are referencing the same vulnerabilities in their reports, and it may take extra time and resources to resolve these discrepancies and correlate the information. This lack of interoperability can cause delays and inconsistencies in security assessment, reporting, decision-making, and vulnerability remediation, as well as hampering communications both within organizations and between organizations. Use of standardized names also helps minimize confusion regarding which problem is being addressed, such as which vulnerabilities a new patch mitigates. This helps organizations to quickly identify the information they need, such as remediation information, when a new problem arises. This publication provides information and recommendations related to two commonly used vulnerability naming schemes: Common Vulnerabilities and Exposures (CVE), and Common Configuration Enumeration (CCE). Both are described in detail below.

Common Vulnerabilities and Exposures (CVE)

The CVE vulnerability naming scheme is for a dictionary of unique, common names for publicly known software flaws. CVE provides the following:

- < A comprehensive list of publicly known software flaws
- < A globally unique name to identify each vulnerability
- < A basis for discussing priorities and risks of vulnerabilities
- < A way for a user of disparate products and services to integrate vulnerability information

A CVE vulnerability entry consists of a unique identifier number.

f) ASSIGNING CWE CODES TO EACH VULNERABILITY:

The main goal of the CWE initiative is to stop vulnerabilities at the source by educating software and hardware acquirers, architects, designers, and programmers on how to eliminate the most common mistakes before a product is delivered.

CWE serves as a resource for programmers as they write code, for architects as they design new software, for hardware engineers as they create physical components, and supports educators in teaching security as part of curriculum for software and hardware engineering, computer science, and management information systems; CWE ultimately helps them prevent the kinds of security vulnerabilities that have plagued the software and hardware industries and put enterprises at risk. CWE continues to evolve as a collaborative community effort to populate a publicly available repository of software and hardware errors in code, design, architecture, and implementation for developers and security practitioners that can also be utilized by tool vendors for tagging what their tool's report and claim to cover.

CWE is industry-endorsed by the [CWE Community](#), which includes representatives from major operating systems vendors, commercial information security tool vendors, academia, government agencies, and research institutions. Community members participate in the development of CWE on the [CWE Community Research](#) email list. This means the [CWE List](#), as well as its follow-on [CWSS](#) and [CWRAF](#) efforts, reflect the insights and combined expertise of the broadest possible collection of

information technology and information security professionals.

Assigning Common Weakness Enumeration (CWE) codes to each vulnerability is an important step in the vulnerability assessment process. CWE is a community-developed list of common software and hardware weaknesses, maintained by the MITRE Corporation, which provides a common language for identifying, understanding, and mitigating software vulnerabilities. By assigning a CWE code to each vulnerability, security professionals and developers can better understand the nature of the vulnerability and take appropriate steps to mitigate the risk.

Targeted at both the development community and the community of security practitioners, Common Weakness Enumeration (CWE™) is a formal list or dictionary of common software and hardware weaknesses that can occur in architecture, design, code, or implementation that can lead to exploitable security vulnerabilities. CWE was created to serve as a common language for describing security weaknesses; serve as a standard measuring stick for security tools targeting these weaknesses; and to provide a common baseline standard for weakness identification, mitigation, and prevention efforts.

A “weakness” is a condition in a software, firmware, hardware, or service component that, under certain circumstances, could contribute to the introduction of vulnerabilities.

g) PROVIDING OWASP CATEGORY AND DESCRIPTION FOR EACH VULNERABILITY:

Providing OWASP category and description for each vulnerability involves categorizing the vulnerabilities based on the OWASP Top 10, which is a list of the most common web application vulnerabilities. This process involves identifying which OWASP category the vulnerability falls under and providing a detailed description of the vulnerability. This information is important because it helps developers and security professionals prioritize which vulnerabilities to address first, based on their potential impact on the application's security.

- Lack of input validation on user input
- Lack of sufficient logging mechanism
- Fail-open error handling
- Not closing the database connection properly

For a great overview, check out the [OWASP Top Ten Project](#). You can read about the top vulnerabilities and download a paper that covers them in detail. Many organizations and agencies use the Top Ten as a way of creating awareness about application security.

NOTE: Before you add a vulnerability, please search and make sure there isn't an equivalent one already. You may want to consider creating a redirect if the topic is the same. Every vulnerability article has a defined structure.

3) BUSSINESS IMPACT ASSESSMENT:

A business impact analysis (BIA) is the process of determining the criticality of business activities and associated resource requirements to ensure operational resilience and continuity of operations during and after a business disruption. The BIA quantifies the impacts of disruptions on service delivery, risks to service delivery, and recovery time objectives (RTOs) and recovery point objectives (RPOs). These recovery requirements are then used to develop strategies, solutions and plans.

a)CONDUCT A THOROUGH ANALYSIS OF THE POTENTIAL BUSINESS IMPACT OF EACH VULNERABILITY:

securing approval for the BIA from senior management;
gathering trained people who can perform a BIA;
preparing a BIA plan;
gathering information from questionnaires, interviews and documentation that is relevant to the analysis;
evaluating the collected information and interview data

Conducting a business impact assessment is an important step in the vulnerability identification and reporting process. This involves analyzing the potential impact that each vulnerability could have on the organization's operations, reputation, and finances. The assessment should take into account the likelihood of the vulnerability being exploited, the potential damage that could be caused, and

the organization's ability to respond and recover from such an incident. By conducting a thorough business impact assessment, stakeholders can prioritize the vulnerabilities and allocate resources appropriately to mitigate the risks.

b) UNDERSTAND THE POTENTIAL CONSEQUENCES OF EACH VULNERABILITY ON THE BUSINESS:

BIAs have many goals. They are used to determine the most crucial business functions, systems, staff and technology resources needed for operations to run optimally. They are also used to assess the time frame within which the functions must be recovered for the organization to restore operations to a normal working state. The analysis may be manual or computer-assisted. Challenges include determining the revenue impact of a business function and quantifying the long-term impact of losses in market share, business reputation or customers. Impacts to consider include the following:

- delayed sales or income
- increased labor expenses
- regulatory fines
- contractual penalties
- customer dissatisfaction

The business impact analysis report typically includes an executive summary, information on the methodology for data

gathering and analysis, detailed findings on the various business units and functional areas, charts and diagrams to illustrate potential losses, and recommendations for recovery.

c) CONDUCTING A BUSINESS IMPACT ASSESSMENT:

Conducting a business impact assessment involves evaluating the potential impact of vulnerabilities on the business. This involves identifying critical business processes and assessing the impact of the vulnerabilities on these processes. By conducting a business impact assessment, organizations can prioritize vulnerabilities based on their potential impact on the business.

d) UNDERSTANDING POTENTIAL CONSEQUENCES OF VULNERABILITIES:

Understanding potential consequences of vulnerabilities is crucial in determining the level of risk posed by each vulnerability. This involves assessing the likelihood of a vulnerability being exploited, the potential impact of an exploit, and the potential consequences of a successful attack. By understanding the potential consequences of vulnerabilities, organizations can develop appropriate mitigation strategies to minimize the risk to the business.

e) ASSESSING THE RISK TO THE BUSINESS:

Assessing the risk to the business involves evaluating the likelihood of a vulnerability being exploited and the potential impact it could have on the organization. The risk assessment should take into account factors such as the threat landscape, the value of the assets at risk, and the organization's current security posture. By conducting a risk assessment, stakeholders can identify vulnerabilities that pose the greatest risk to the organization and prioritize their remediation efforts. It is important to conduct ongoing risk assessments to ensure that vulnerabilities are identified and addressed in a timely manner.

4) VULNERABILITY PATH AND PARAMETER

IDENTIFICATION:

a) METHODS FOR IDENTIFYING VULNERABILITY

PATH AND PARAMETERS:

To scan target website vulnerabilities there many tools in kali linux.

- Nikto.
- Nmap.
- Brup suite.
- Dbscan.etc.....

According to project to scan vulnerability of target we used tool called Nikto and Nmap.

- Nikto and parameter of nikto.

Nikto is an Open Source software written in Perl language that is used to scan a web-server for the vulnerability that can be exploited and can compromise the server. It can also check for outdated version details of 1200 server and can detect problems with specific version details of over 200 servers

Nmap and parameter of nmap:

```

root@kali:~# nikto -h

Options:
  -ask+           Whether to ask about submitting updates
                  yes   Ask about each (default)
                  no    Don't ask, don't send
                  auto  Don't ask, just send
  -check6          Check if IPv6 is working (connects to ipv6.google.com or va
  -Cgidirs+        Scan these CGI dirs: "none", "all", or values like "/cgi /
  -config+         Use this config file
  -Display+        Turn on/off display outputs:
                  1    Show redirects
                  2    Show cookies received
                  3    Show all 200/OK responses
                  4    Show URLs which require authentication
                  D    Debug output
                  E    Display all HTTP errors
                  P    Print progress to STDOUT
                  S    Scrub output of IPs and hostnames
                  V    Verbose output
  -dbcheck          Check database and other key files for syntax errors
  -evasion+        Encoding technique:
                  1    Random IRT encoding (non-IIFR)

root@kali:~# nmap -h
Nmap 7.93 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/SX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
  Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
  --exclude-ports <port ranges>: Exclude the specified ports from scanning
  -F: Fast mode - Scan fewer ports than the default scan

```

b) TYPES OF VULNERABILITY PATHS AND PARAMETERS:

In order to understand vulnerability, let us look at some of the main reasons behind it. These have come to light through research experiences and empirical evidence from different parts of the world. Some of the key reasons for ever-increasing levels of vulnerability are: Rapid population

growth where disaster events can claim more lives. Environmental degradation due to poor land use, deforestation, overcultivation and overgrazing. These render the land more prone to floods and landslides. Increased rate of industrialization and rapid urbanization without the necessary safety measures. Impoverished conditions. Blind adherence to cultural practices Gender inequalities. War and civil strife. Lack of public awareness and information.

c) COMMON TOOLS AND TECHNIQUES FOR IDENTIFYING VULNERABILITY PATHS AND PARAMETERS:

Nikto: an application that scans web-based applications and web servers for known bad files that could potentially be dangerous. Other things that it can detect include outdated configs, port scanning, username enumeration and more.

Skipfish: Skipfish is an automated tool that performs reconnaissance tasks on web servers. It generates a sitemap and then recursively probes the site with penetration tests to identify vulnerabilities.

Wapiti: Wapiti is another penetration testing tool that manages to probe common such SQL injection, cross-site scripting and it uses GET and POST methods as part of its attacking capabilities.

OWASP-ZAP: The Zed Attack Proxy scanner is a pentesting app that allows you to test web apps while still in the dev stage. This lets you design tests to find problems before they get released into production environments.

XSSPY: As the name suggests, this is a Python tool that tests for cross-site scripting vulnerabilities in websites and web applications. After an initial scan that maps out the entire site, it then begins the detailed task of scanning every element that it uncovered in search of XSS vulnerabilities.

W3af: This is a web application framework that lets you attack and audit web apps and uncovers and exploits web application vulnerabilities as part of

your vulnerabilities assessment. It is available as a GUI and console application, and it has over 130 different plugins for different tasks.

The most common software security vulnerabilities include:

- Missing data encryption
- OS command injection
- SQL injection
- Buffer overflow
- Missing authentication for critical function

d) BEST PRACTICES FOR VULNERABILITY PATH AND PARAMETER IDENTIFICATION:

To ensure that vulnerability paths and parameters are identified accurately and comprehensively, it is important to use a combination of manual and automated testing methods. It is also important to test applications in different environments and with different user roles to identify all possible attack vectors. Finally, thorough documentation and reporting of identified vulnerabilities and their associated paths and parameters is crucial for developers to be able to address the vulnerabilities effectively.

e) CHALLENGES AND LIMITATIONS OF VULNERABILITY PATH AND PARAMETER IDENTIFICATION:

One of the biggest challenges in identifying vulnerability paths and parameters is the constantly evolving nature of vulnerabilities and attack methods. Additionally, some vulnerabilities may be difficult to identify and require specialized knowledge and skills to detect. Another limitation is the potential for false positives or false negatives in vulnerability scanning and testing, which can lead to wasted time and resources.

5) DETAILED INSTRUCTION FOR VULNERABILITY REPRODUCTION:

This policy applies to the following systems and services:

- Websites operated by us.
- Software applications made available by us for use on or through computers and mobile devices.
- Our social media pages and applications.

Any system or service not expressly listed above, such as any connected systems or services, is excluded from scope and is not authorized for testing. Additionally, vulnerabilities found in systems owned or controlled by third parties fall outside the scope of this policy. They should be reported directly to the third party according to their policies.

Though we develop and maintain other Internet-accessible systems or services, we ask that active research and testing be conducted only on the systems and services covered by the scope of this policy.

a) IMPORTANCE OF PROVIDING DETAILED INSTRUCTIONS:

Providing detailed instructions for reproducing vulnerabilities is crucial for developers to understand the specific steps required to fix the vulnerability. Without detailed instructions, developers may have difficulty understanding the nature of the vulnerability and how to fix it. Detailed instructions also ensure that vulnerabilities are correctly identified and addressed, reducing the risk of future attacks.

Giving clear instructions to students can ensure that they fully comprehend what they need to do to achieve in your classroom. It will ease students' nerves, assuage their insecurities, and help them confirm your expectations so that they can be happy and successful in school.

b) COMPONENTS OF A WELL-WRITTEN VULNERABILITY REPRODUCTION INSTRUCTION:

A well-written vulnerability reproduction instruction should include a detailed description of the vulnerability, steps to reproduce the

vulnerability, and expected outcomes. The instruction should also include information on the platform or application affected, the severity of the vulnerability, and any potential impact of the business.

An essential skill for a security researcher is the ability to write concise and clear vulnerability reports. A well-written vulnerability report will help the security team reproduce and fix the issue faster and minimize the possibility of exploitation. In this post, we are going to go into the components of a good vulnerability report and some tips and tricks I've learned along the way.

c) STEPS FOR REPRODUCING VULNERABILITIES:

The steps for reproducing vulnerabilities typically involve a series of actions or inputs that trigger the vulnerability. These steps must be clearly defined and detailed to ensure that developers can understand and replicate the vulnerability. Additionally, steps for reproducing vulnerabilities should be consistent across multiple systems or environments to ensure that the vulnerability can be identified and addressed in a timely manner.

d) BEST PRACTICES FOR WRITING EFFECTIVE VULNERABILITY REPRODUCTION INSTRUCTIONS:

Effective vulnerability reproduction instructions should be clear, concise, and easy to understand. Instructions should be written in plain language and avoid technical jargon. Screenshots or videos can be used to supplement written instructions and provide visual aids for developers.

There are five main stages in the vulnerability management cycle include:

- Step 1. Assess
- Step 2. Prioritize
- Step 3. Act
- Step 4. Reassess
- Step 5. Improve

e) TOOLS AND TECHNIQUES FOR VERIFYING VULNERABILITY FIXES:

Tools and techniques for verifying vulnerability fixes may include automated testing tools, manual testing, and code reviews. These methods can be used to ensure that vulnerabilities have been successfully fixed and that no new vulnerabilities have been introduced.

Vulnerability scanning or vulnerability assessment is a systematic process of finding security loopholes in any system addressing potential vulnerabilities.

The purpose of vulnerability assessments is to prevent the possibility of unauthorized access to systems. Vulnerability testing preserves the confidentiality, integrity, and availability of the system. The system refers to any computers, networks, network devices, software, web application, cloud computing, etc.

f) CHALLENGES AND LIMITATIONS OF VULNERABILITY REPRODUCTION INSTRUCTION:

Challenges and limitations of vulnerability reproduction instruction may include differences in system configurations or environments, difficulty in replicating complex vulnerabilities, and the need for access to source code or proprietary systems. It is important to address these challenges to ensure that vulnerabilities are accurately identified and addressed.

Cyber threats and attacks are on the rise for industrial, manufacturing and critical infrastructure organizations. Many of these threats are the result of vulnerabilities present in the organization's OT systems. Targeted threat actors or untargeted ransomware attacks can exploit these vulnerabilities to gain access into industrial networks for financial gain or to interrupt operations.

The rise of cyber threats are due to factors such as the acceleration of IIoT technology and digital transformation such as Industry 4.0 in manufacturing, an increase in remote work

and remote access amid the COVID-19 pandemic, and the prevalence of ransomware.

6) COMPREHENSIVE AND DETAILED REPORTING:

By definition, a comprehensive report is intended to explore a topic or an idea in great detail. In business, comprehensive reports are often used to evaluate and discuss a company's financial situation. Comprehensive reports may be used for other purposes as well, such as summarizing a new business trend or describing a new target market. Learning exactly how to write a comprehensive report can be a useful business skill for employees at any level.

a) IMPORTANCE OF COMPREHENSIVE AND DETAILED REPORTING:

Comprehensive and detailed reporting is crucial for businesses and organizations to make informed decisions. It involves analyzing and presenting data in a clear and concise manner, which helps stakeholders to identify patterns, trends, and potential problems. Comprehensive and detailed reporting provides an accurate picture of an organization's operations, financial performance, and overall health, which can be used to guide strategic planning and resource allocation. The Importance of Reporting

The importance of reporting can be understood mainly by an experienced project manager. If you cannot properly account for occurrences within your organization, then you cannot call yourself a project manager. The purpose of project

management is to be able to comprehend, comply, conform, control, and communicate in order to succeed with a company's goals. As a project manager, none of these aspects can be achieved without proper reporting.

b) KEY COMPONENTS OF COMPREHENSIVE AND DETAILED REPORTING:

Key components of comprehensive and detailed reporting include accurate and relevant data, effective data analysis, clear and concise presentation of findings, and actionable recommendations. The report should be well-organized, easy to understand, and tailored to the audience's needs. It should also provide context for the data presented, such as benchmarking against industry standards or historical data. The first thing you need to do is identify your audience and what they need to know. Then think through what the report is about—what information needs to be in it, what information is best left out?

Also, remember that the key difference between an essay and a report is that an essay focuses on developing an argument or point of view, while a report centres on solving a problem and presenting research findings. You can also use headings (this is something you wouldn't do in an essay) to identify sections of the report (i.e., Discussion, Conclusion, etc.).

c) STRATEGIES FOR EFFECTIVE REPORTING:

Strategies for effective reporting include identifying the purpose and scope of the report, understanding the audience's needs, selecting appropriate data sources and analysis techniques, and using clear and concise language to present findings. The report should be well-organized, visually

appealing, and use data visualization tools to help the audience better understand the data.

d) CHALLENGES IN IMPLEMENTING COMPREHENSIVE AND DETAILED REPORTING:

Challenges in implementing comprehensive and detailed reporting include data quality issues, data silos, lack of resources, and difficulty in identifying the right metrics to measure. Organizations also face challenges in presenting data in a way that is easily digestible for different stakeholders, such as executives, managers, and frontline employees.

e) IMPACT OF COMPREHENSIVE AND DETAILED REPORTING ON DECISION MAKING:

Comprehensive and detailed reporting can have a significant impact on decision-making by providing stakeholders with the information they need to make informed decisions. It can help identify areas for improvement, highlight potential risks, and guide resource allocation. By providing a comprehensive view of an organization's operations, financial performance, and overall health, stakeholders can make more informed decisions that align with their strategic goals.

f) BEST PRACTICES FOR CREATING COMPREHENSIVE AND DETAILED REPORTS:

Best practices for creating comprehensive and detailed reports include defining the purpose and scope of the report, selecting appropriate data sources and analysis techniques, using clear and concise language to present findings, and including actionable recommendations. The report should be well-organized, visually appealing, and tailored to the audience's needs. It should also provide context for the data presented, such as benchmarking against industry standards or historical data.

CONCLUSION

In conclusion, Network Vulnerability Assessments (NVA) is how you determine whether your network is secure against malicious attacks. It involves identifying possible weaknesses in your network and then taking steps to fix them. The goal of any Vulnerability is to ensure that no single point of failure exists on your network.

You can conduct an Network Vulnerability using many different approaches, and each process has its strengths and weaknesses. Therefore, it's important to choose the right method based on your needs.