January 3, 2025

Vulnerability Scan
Report

Prepared By

Vulnsync

Overview

1 Executive Summary

Vulnerability scans were conducted on select servers, networks, websites, and applications. This report contains the discovered potential vulnerabilities from these scans. Vulnerabilities have been classified by severity. Higher severity indicates a greater risk of a data breach, loss of integrity, or availability of the targets.

1.1 Total Vulnerabilities

Below are the total number of vulnerabilities found by severity. Critical vulnerabilities are the most severe and should be evaluated first. An accepted vulnerability is one which has been manually reviewed and classified as acceptable to not fix at this time, such as a false positive detection or an intentional part of the system's architecture.

Critical	High	Medium	Low	Accepted
0	0	8	9	0

1.2 Report Coverage

This report includes findings for 1 target scanned. Each target is a single URL, IP address, or fully qualified domain name (FQDN).

Vulnerability Categories

5 Active Web Application Vulnerabilities	5 Passive Web Application Vulnerabilities	3 Network Vulnerabilities
2 SSL/TLS Security	2 Open TCP Ports	0 Open UDP Ports

2 Vulnerabilities By Target

This section contains the vulnerability findings for each scanned target. Prioritization should be given to the targets with the highest severity vulnerabilities. However, it is important to take into account the purpose of each system and consider the potential impact a breach or an outage would have for the particular target.

2.1 Targets Summary

The number of potential vulnerabilities found for each target by severity.

Target	Critical	High	Medium	Low	Accepted
example.com	0	0	8	9	0

2.2 Target Breakdowns

Details for the potential vulnerabilities found for each target by scan type.

example.com

Total Risks

0	0	8	9	0

47% 53%

Active Web Application Vulnerabilities	Severity	First Detected	Last Detected
Missing Anti-clickjacking Header	Medium	0 days ago	0 days ago
Content Security Policy (CSP) Header Not Set	Medium	0 days ago	0 days ago
X-Content-Type-Options Header Missing	Low	0 days ago	0 days ago
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	0 days ago	0 days ago
Strict-Transport-Security Header Not Set	Low	0 days ago	0 days ago

Passive Web Application Vulnerabilities	Severity	First Detected	Last Detected
Missing Anti-clickjacking Header	Medium	0 days ago	0 days ago
Content Security Policy (CSP) Header Not Set	Medium	0 days ago	0 days ago
X-Content-Type-Options Header Missing	Low	0 days ago	0 days ago
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	0 days ago	0 days ago
Strict-Transport-Security Header Not Set	Low	0 days ago	0 days ago

Network Vulnerabilities	Severity	First Detected	Last Detected
SSL/TLS: Report Weak Cipher Suites cvss score: 5.9	Medium	0 days ago	0 days ago

SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection cvss score: 4.3	Medium	0 days ago	0 days ago
TCP Timestamps Information Disclosure cvss score: 2.6	Low	0 days ago	0 days ago

SSL/TLS Security	Severity	First Detected	Last Detected
TLS 1.1 is considered an insecure encryption protocol and should be disabled.	Medium	0 days ago	0 days ago
TLS 1.0 is considered an insecure encryption protocol and should be disabled.	Medium	0 days ago	0 days ago

Open TCP Ports	Severity	First Detected	Last Detected
Open TCP Port: 443	Low	0 days ago	0 days ago
Open TCP Port: 80	Low	0 days ago	0 days ago

3 Active Web Application Vulnerabilities

The OWASP ZAP Active Web Application scan crawls the pages of a website or web application testing for vulnerabilities and configuration weaknesses. The active scan includes all of the passive scan tests and additionally makes requests and submits forms to actively test an application for more vulnerabilities. The active scan tests for vulnerabilities such as SQL injection, remote command execution, XSS, and more.

3.1 Total Vulnerabilities

Total number of vulnerabilities found by severity.

Critical	High	Medium	Low	Accepted
0	0	2	3	0

3.2 Vulnerabilities Breakdown

Summary list of all detected vulnerabilities.

Title	Severity	Open	Accepted
Missing Anti-clickjacking Header	Medium	1	0
Content Security Policy (CSP) Header Not Set	Medium	1	0
X-Content-Type-Options Header Missing	Low	1	0
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	1	0
Strict-Transport-Security Header Not Set	Low	1	0

3.3 Vulnerability Details

Detailed information about each potential vulnerability found by the scan.

Missing Anti-clickjacking Header

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Medium	1 target	0 days ago

Description

The response does not protect against 'ClickJacking' attacks. It should include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options.

Solution

Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app.

If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.

Instances (1 of 1)

uri: https://example.com/
method: GET
param: x-frame-options

References

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

Vulnerable Target	First Detected	Last Detected
example.com	0 days ago	0 days ago

Content Security Policy (CSP) Header Not Set

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Medium	1 target	0 days ago

Description

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.

Solution

Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Instances (1 of 3)

uri: https://example.com/
method: GET

References

https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy
https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
https://www.w3.org/TR/CSP/
https://w3c.github.io/webappsec-csp/
https://web.dev/articles/csp
https://caniuse.com/#feat=contentsecuritypolicy
https://content-security-policy.com/

Vulnerable Target	First Detected	Last Detected
example.com	0 days ago	0 days ago

X-Content-Type-Options Header Missing

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Low	1 target	0 days ago

Description

The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.

Solution

Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.

If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.

Instances (1 of 1)

uri: https://example.com/
method: GET
param: x-content-type-options
otherinfo: This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scan rule will not alert on client or server error responses.

References

https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/compatibility/gg622941(v=vs.85)
https://owasp.org/www-community/Security_Headers

Vulnerable Target	First Detected	Last Detected
example.com	0 days ago	0 days ago

Server Leaks Version Information via "Server" HTTP Response Header Field

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Low	1 target	0 days ago

Description

The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.

Solution

Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Instances (1 of 4)

uri: https://example.com/
method: GET
evidence: ECAcc (agb/534E)

References

https://httpd.apache.org/docs/current/mod/core.html#servertokens
https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ff648552(v=pandp.10)
https://www.troyhunt.com/shhh-dont-let-your-response-headers/

Vulnerable Target	First Detected	Last Detected
example.com	0 days ago	0 days ago

Strict-Transport-Security Header Not Set

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Low	1 target	0 days ago

Description

HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure HTTPS connections (i.e. HTTP layered over TLS/SSL). HSTS is an IETF standards track protocol and is specified in RFC 6797.

Solution

Ensure that your web server, application server, load balancer, etc. is configured to enforce Strict-Transport-Security.

Instances (1 of 3)

uri: https://example.com/
method: GET

References

https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html
https://owasp.org/www-community/Security_Headers
https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security
https://caniuse.com/stricttransportsecurity
https://datatracker.ietf.org/doc/html/rfc6797

Vulnerable Target	First Detected	Last Detected
example.com	0 days ago	0 days ago

4 Passive Web Application Vulnerabilities

The OWASP ZAP Passive Web Application scan crawls the pages of a website or web application. The passive scan inspects each page as well as the requests and responses sent between the server. The passive scan checks for vulnerabilities such as cross-domain misconfigurations, insecure cookies, vulnerable Javascript dependencies, and more.

4.1 Total Vulnerabilities

Total number of vulnerabilities found by severity.

Critical	High	Medium	Low	Accepted
0	0	2	3	0

4.2 Vulnerabilities Breakdown

Summary list of all detected vulnerabilities.

Title	Severity	Open	Accepted
Missing Anti-clickjacking Header	Medium	1	0
Content Security Policy (CSP) Header Not Set	Medium	1	0
X-Content-Type-Options Header Missing	Low	1	0
Server Leaks Version Information via "Server" HTTP Response Header Field	Low	1	0
Strict-Transport-Security Header Not Set	Low	1	0

4.3 Vulnerability Details

Detailed information about each potential vulnerability found by the scan.

Missing Anti-clickjacking Header

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Medium	1 target	0 days ago

Description

The response does not protect against 'ClickJacking' attacks. It should include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options.

Solution

Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app.

If you expect the page to be framed only by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page to be framed, you should use DENY. Alternatively consider implementing Content Security Policy's "frame-ancestors" directive.

Instances (1 of 1)

uri: https://example.com/
method: GET
param: x-frame-options

References

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

Vulnerable Target	First Detected	Last Detected
example.com	0 days ago	0 days ago

Content Security Policy (CSP) Header Not Set

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Medium	1 target	0 days ago

Description

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.

Solution

Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Instances (1 of 3)

uri: https://example.com/
method: GET

References

https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy
https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
https://www.w3.org/TR/CSP/
https://w3c.github.io/webappsec-csp/
https://web.dev/articles/csp
https://caniuse.com/#feat=contentsecuritypolicy
https://content-security-policy.com/

Vulnerable Target	First Detected	Last Detected
example.com	0 days ago	0 days ago

X-Content-Type-Options Header Missing

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Low	1 target	0 days ago

Description

The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.

Solution

Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.

If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.

Instances (1 of 1)

uri: https://example.com/
method: GET
param: x-content-type-options
otherinfo: This issue still applies to error type pages (401, 403, 500, etc.) as those pages are often still affected by injection issues, in which case there is still concern for browsers sniffing pages away from their actual content type. At "High" threshold this scan rule will not alert on client or server error responses.

References

https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/compatibility/gg622941(v=vs.85)
https://owasp.org/www-community/Security_Headers

Vulnerable Target	First Detected	Last Detected
example.com	0 days ago	0 days ago

Server Leaks Version Information via "Server" HTTP Response Header Field

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Low	1 target	0 days ago

Description

The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.

Solution

Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Instances (1 of 4)

uri: https://example.com/
method: GET
evidence: ECAcc (nyd/D111)

References

https://httpd.apache.org/docs/current/mod/core.html#servertokens
https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ff648552(v=pandp.10)
https://www.troyhunt.com/shhh-dont-let-your-response-headers/

Vulnerable Target	First Detected	Last Detected
example.com	0 days ago	0 days ago

Strict-Transport-Security Header Not Set

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Low	1 target	0 days ago

Description

HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure HTTPS connections (i.e. HTTP layered over TLS/SSL). HSTS is an IETF standards track protocol and is specified in RFC 6797.

Solution

Ensure that your web server, application server, load balancer, etc. is configured to enforce Strict-Transport-Security.

Instances (1 of 3)

uri: https://example.com/
method: GET

References

https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html
https://owasp.org/www-community/Security_Headers
https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security
https://caniuse.com/stricttransportsecurity
https://datatracker.ietf.org/doc/html/rfc6797

Vulnerable Target	First Detected	Last Detected
example.com	0 days ago	0 days ago

5 SSL/TLS Security

The SSLyze security scan tests for misconfigured SSL/TLS certificates, expired certificates, weak ciphers, and SSL/TLS vulnerabilities such as Heartbleed.

5.1 Total Vulnerabilities

Total number of vulnerabilities found by severity.

Critical	High	Medium	Low	Accepted
0	0	2	0	0

5.2 Vulnerabilities Breakdown

Summary list of all detected vulnerabilities.

Title	Severity	Open	Accepted
TLS 1.1 is considered an insecure encryption protocol and should be disabled.	Medium	1	0
TLS 1.0 is considered an insecure encryption protocol and should be disabled.	Medium	1	0

5.3 Vulnerability Details

Detailed information about each potential vulnerability found by the scan.

TLS 1.1 is considered an insecure encryption protocol and should be disabled.

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Medium	1 target	0 days ago

Vulnerable Target	First Detected	Last Detected
example.com	0 days ago	0 days ago

TLS 1.0 is considered an insecure encryption protocol and should be disabled.

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Medium	1 target	0 days ago

Vulnerable Target	First Detected	Last Detected
example.com	0 days ago	0 days ago

6 Network Vulnerabilities

The OpenVAS network vulnerability scan tests servers and internet connected devices for over 150,000 vulnerabilities. OpenVAS uses the Common Vulnerability Scoring System (CVSS) to quantify the severity of findings. 0.0 is the lowest severity and 10.0 is the highest.

6.1 Total Vulnerabilities

Total number of vulnerabilities found by severity.

Critical	High	Medium	Low	Accepted
0	0	2	1	0

6.2 Vulnerabilities Breakdown

Summary list of all detected vulnerabilities.

Title	Severity	CVSS Score	Open	Accepted
SSL/TLS: Report Weak Cipher Suites	Medium	5.9	1	0
SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection	Medium	4.3	1	0
TCP Timestamps Information Disclosure	Low	2.6	1	0

6.3 Vulnerability Details

Detailed information about each potential vulnerability found by the scan.

SSL/TLS: Report Weak Cipher Suites

SEVERITY	AFFECTED TARGETS	LAST DETECTED	CVSS SCORE
Medium	1 target	0 days ago	5.9

Description

This routine reports all Weak SSL/TLS cipher suites accepted by a service.

NOTE: No severity for SMTP services with 'Opportunistic TLS' and weak cipher suites on port 25/tcp is reported. If too strong cipher suites are configured for this service the alternative would be to fall back to an even more insecure cleartext communication.

These rules are applied for the evaluation of the cryptographic strength:

- RC4 is considered to be weak (CVE-2013-2566, CVE-2015-2808)

- Ciphers using 64 bit or less are considered to be vulnerable to brute force methods and therefore considered as weak (CVE-2015-4000)

- 1024 bit RSA authentication is considered to be insecure and therefore as weak

- Any cipher considered to be secure for only the next 10 years is considered as medium

- Any other cipher is considered as strong

Solution

The configuration of this services should be changed so that it does not accept the listed weak cipher suites anymore.

Please see the references for more resources supporting you with this task.

References

CVE-2013-2566
CVE-2015-2808
CVE-2015-4000
https://www.bsi.bund.de/SharedDocs/Warnmeldungen/DE/CB/warnmeldung_cb-k16-1465_update_6.html
https://bettercrypto.org/
https://mozilla.github.io/server-side-tls/ssl-config-generator/

Vulnerable Target	First Detected	Last Detected
example.com	0 days ago	0 days ago

SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection

SEVERITY	AFFECTED TARGETS	LAST DETECTED	CVSS SCORE
Medium	1 target	0 days ago	4.3

Description

It was possible to detect the usage of the deprecated TLSv1.0 and/or TLSv1.1 protocol on this system.

The TLSv1.0 and TLSv1.1 protocols contain known cryptographic flaws like:

- CVE-2011-3389: Browser Exploit Against SSL/TLS (BEAST)

- CVE-2015-0204: Factoring Attack on RSA-EXPORT Keys Padding Oracle On Downgraded Legacy Encryption (FREAK)

An attacker might be able to use the known cryptographic flaws to eavesdrop the connection between clients and the service to get access to sensitive data transferred within the secured connection.

Furthermore newly uncovered vulnerabilities in this protocols won't receive security updates anymore.

Solution

It is recommended to disable the deprecated TLSv1.0 and/or TLSv1.1 protocols in favor of the TLSv1.2+ protocols. Please see the references for more information.

References

CVE-2011-3389
CVE-2015-0204
https://ssl-config.mozilla.org/
https://bettercrypto.org/
https://datatracker.ietf.org/doc/rfc8996/
https://vnhacker.blogspot.com/2011/09/beast.html
https://web.archive.org/web/20201108095603/https://censys.io/blog/freak
https://www.enisa.europa.eu/publications/algorithms-key-size-and-parameters-report-2014

Vulnerable Target	First Detected	Last Detected
example.com	0 days ago	0 days ago

TCP Timestamps Information Disclosure

SEVERITY	AFFECTED TARGETS	LAST DETECTED	CVSS SCORE
Low	1 target	0 days ago	2.6

Description

The remote host implements TCP timestamps and therefore allows to compute the uptime.

The remote host implements TCP timestamps, as defined by RFC1323/RFC7323.

A side effect of this feature is that the uptime of the remote host can sometimes be computed.

Solution

To disable TCP timestamps on linux add the line 'net.ipv4.tcp_timestamps

References

https://datatracker.ietf.org/doc/html/rfc1323
https://datatracker.ietf.org/doc/html/rfc7323
https://web.archive.org/web/20151213072445/http://www.microsoft.com/en-us/download/details.aspx?id=9152
https://www.fortiguard.com/psirt/FG-IR-16-090

Vulnerable Target	First Detected	Last Detected
example.com	0 days ago	0 days ago

7 Open TCP Ports

The NMAP TCP port scan discovers open TCP ports with a complete scan of ports 0 to 65535.

7.1 Total Vulnerabilities

Total number of vulnerabilities found by severity.

Critical	High	Medium	Low	Accepted
0	0	0	2	0

7.2 Vulnerabilities Breakdown

Summary list of all detected vulnerabilities.

Title	Severity	Open	Accepted
Open TCP Port: 443	Low	1	0
Open TCP Port: 80	Low	1	0

7.3 Vulnerability Details

Detailed information about each potential vulnerability found by the scan.

Open TCP Port: 443

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Low	1 target	0 days ago

Description

An open port may be an expected configuration. For example, web servers use port 80 to serve websites over http and port 443 to serve websites over https. For a list of commonly used ports see https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers.

An unexpected open port could give unintended access to applications, data, and private networks. Open ports can also be dangerous when expected services are out of date and exploited through security vulnerabilities.

Vulnerable Target	First Detected	Last Detected
example.com	0 days ago	0 days ago

Open TCP Port: 80

SEVERITY	AFFECTED TARGETS	LAST DETECTED
Low	1 target	0 days ago

Description

An open port may be an expected configuration. For example, web servers use port 80 to serve websites over http and port 443 to serve websites over https. For a list of commonly used ports see https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers.

An unexpected open port could give unintended access to applications, data, and private networks. Open ports can also be dangerous when expected services are out of date and exploited through security vulnerabilities.

Vulnerable Target	First Detected	Last Detected
example.com	0 days ago	0 days ago

8 Open UDP Ports

The NMAP UDP port scan discovers open ports of common UDP services

8.1 Total Vulnerabilities

Total number of vulnerabilities found by severity.

Critical	High	Medium	Low	Accepted
0	0	0	0	0

8.2 Vulnerabilities Breakdown

Summary list of all detected vulnerabilities.

Title	Severity	Open	Accepted
No vulnerabilities detected			

9 Glossary

Accepted Vulnerability

An accepted vulnerability is one which has been manually reviewed and classified as acceptable to not fix at this time, such as a false positive scan result or an intentional part of the system's architecture.

Active Web Application Vulnerabilities

The OWASP ZAP Active Web Application scan crawls the pages of a website or web application testing for vulnerabilities and configuration weaknesses. The active scan includes all of the passive scan tests and additionally makes requests and submits forms to actively test an application for more vulnerabilities. The active scan tests for vulnerabilities such as SQL injection, remote command execution, XSS, and more.

Fully Qualified Domain Name (FQDN)

A fully qualified domain name is a complete domain name for a specific website or service on the internet. This includes not only the website or service name, but also the top-level domain name, such as .com, .org, .net, etc. For example, 'www.example.com' is an FQDN.

Passive Web Application Vulnerabilities

The OWASP ZAP Passive Web Application scan crawls the pages of a website or web application. The passive scan inspects each page as well as the requests and responses sent between the server. The passive scan checks for vulnerabilities such as cross-domain misconfigurations, insecure cookies, vulnerable Javascript dependencies, and more.

Network Vulnerabilities

The OpenVAS network vulnerability scan tests servers and internet connected devices for over 150,000 vulnerabilities. OpenVAS uses the Common Vulnerability Scoring System (CVSS) to quantify the severity of findings. 0.0 is the lowest severity and 10.0 is the highest.

Open TCP Ports

The NMAP TCP port scan discovers open TCP ports with a complete scan of ports 0 to 65535.

Open UDP Ports

The NMAP UDP port scan discovers open ports of common UDP services

Vulnerability

A weakness in the computational logic (e.g., code) found in software and hardware components that, when exploited, results in a negative impact to confidentiality, integrity, or availability. Mitigation of the vulnerabilities in this context typically involves coding changes, but could also include specification changes or even specification deprecations (e.g., removal of affected protocols or functionality in their entirety).

SSL/TLS Security

The SSLyze security scan tests for misconfigured SSL/TLS certificates, expired certificates, weak ciphers, and SSL/TLS vulnerabilities such as Heartbleed.

Target

A target represents target is a single URL, IP address, or fully qualified domain name (FQDN) that was scanned.

Severity

Severity represents the estimated impact potential of a particular vulnerability. Severity is divided into 5 categories: Critical, High, Medium, Low and Accepted.

CVSS Score

The CVSS 3.0 score is a global standard for evaluating vulnerabilities with a 0 to 10 scale. CVSS maps to threat levels:
0.1 - 3.9 = Low
4.0 - 6.9 = Medium
7.0 - 8.9 = High
9.0 - 10.0 = Critical