internship studio

ETHICAL HACKING INTERNSHIP

Name: Kalyani Gajanan Lonkar

Institutional Affiliation: Internship Studio

Email: kalyanilonkar5@gmail.com

Task : 2

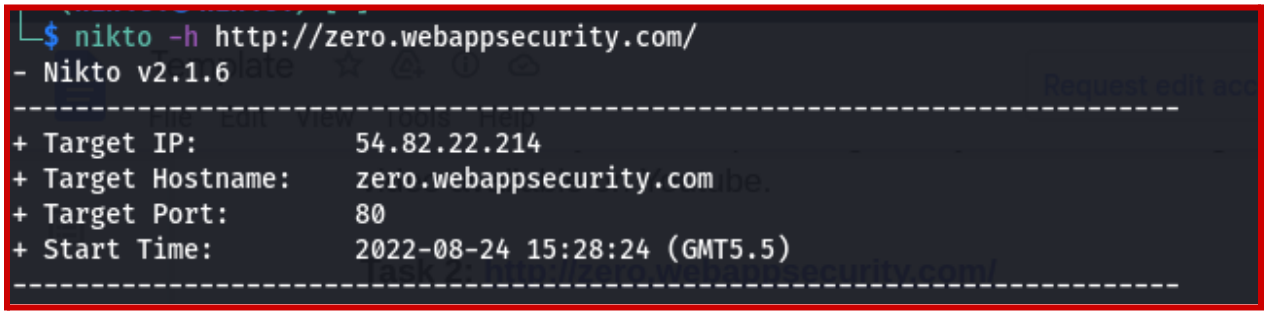
To test the website and find all possible vulnerabilities & loopholes in it.

http://zero.webappsecurity.com/

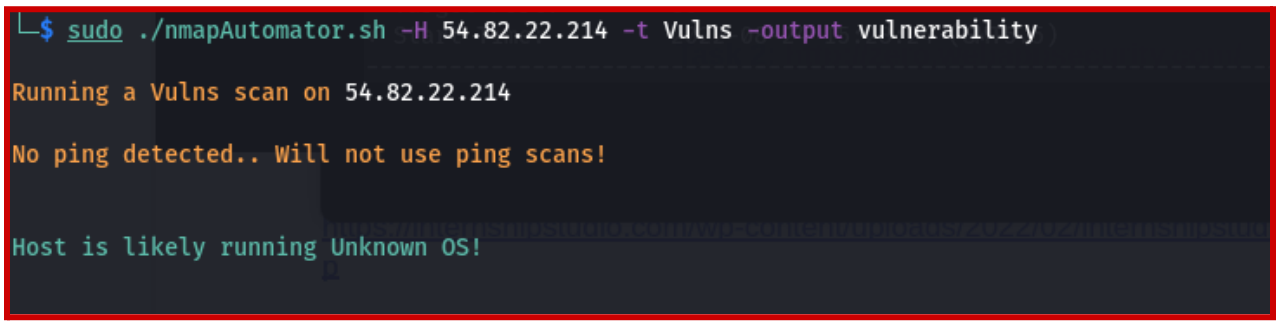
Using the NmapAutomator Vulnerability Scanner:

Firstly I used nikto on the given website, through the nikto scan.

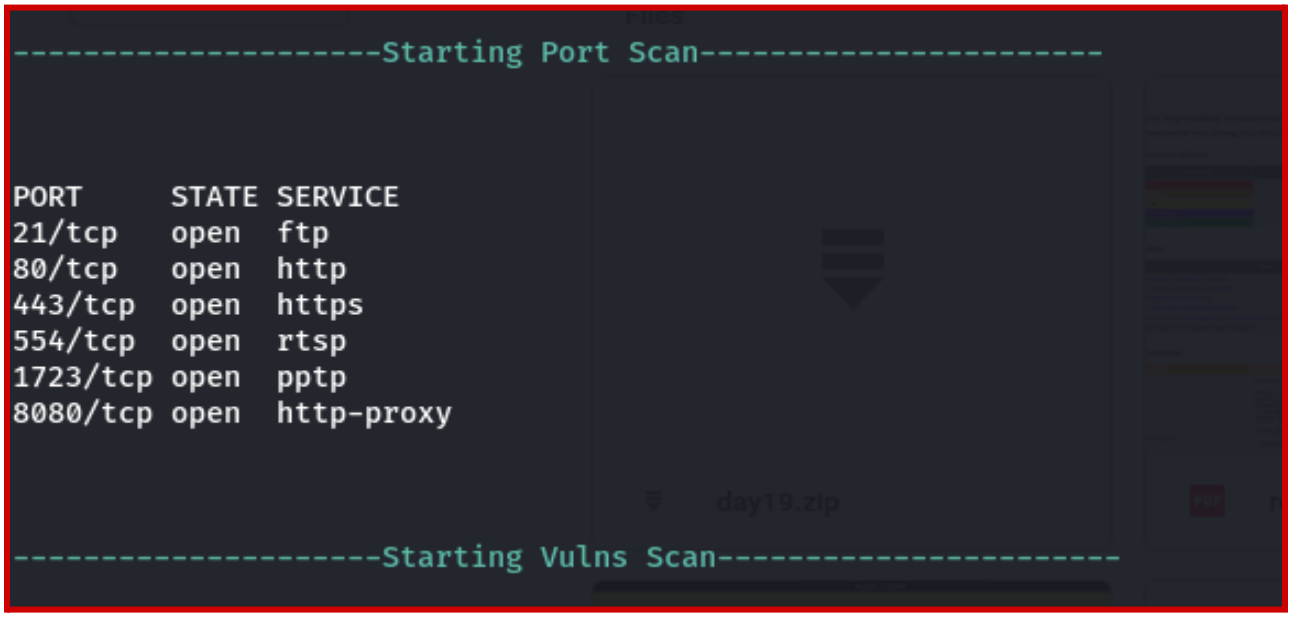
I Got the Target IP 54.82.22.214



Then using the **nmapAutomator** on the websites Ip 54.82.22.214

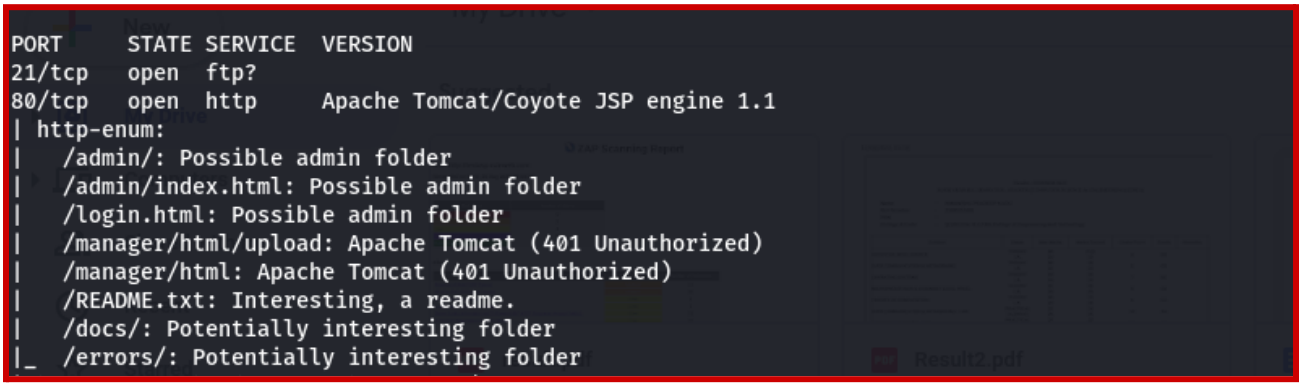


NmapAutomator did Port Scan



NampAutomator did Vulns Scan

Running Vuln Scan on Common Ports We got Vulnerabilities.



1. Vulnerability

ssl-dh-params:
Transport Layer Security (TLS) Protocol DHE_EXPORT Ciphers Downgrade MitM (Logjam)
State : Vulnerable
IDs: CVE:CVE-2015-4000 BID:74733

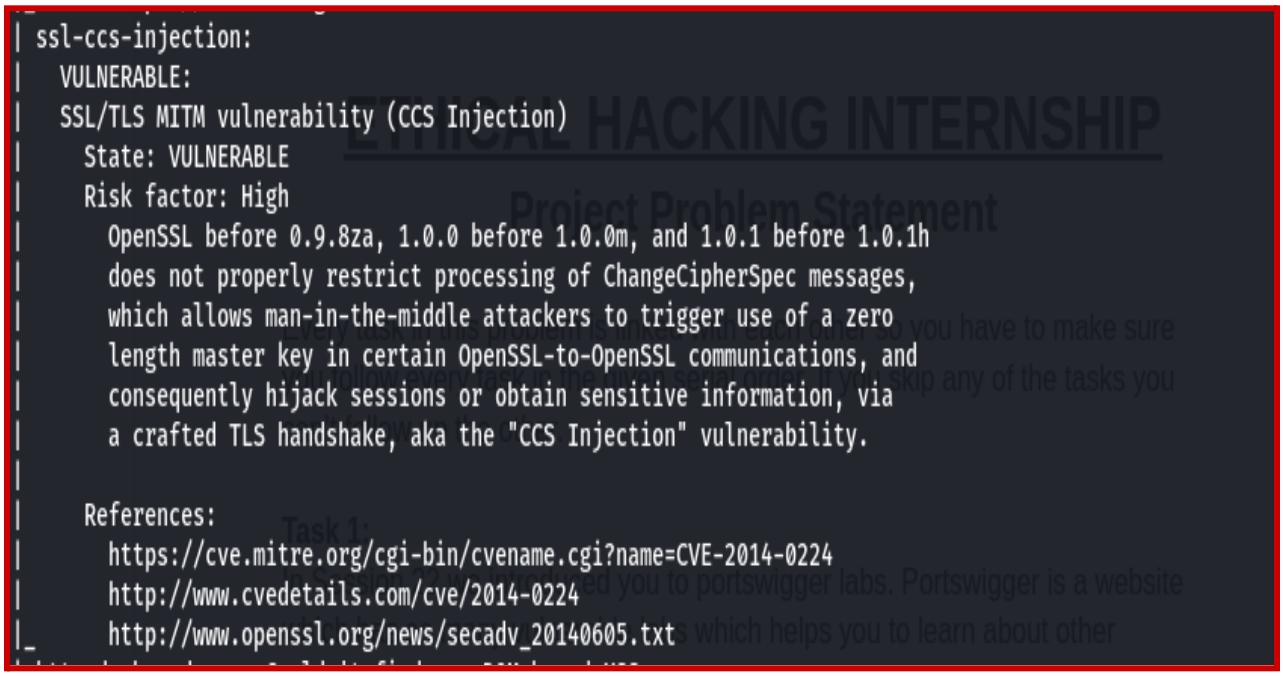
The Transport Layer Security (TLS) protocol contains a flaw that is triggered when handling Diffie-Hellman key exchanges defined with the DHE_EXPORT cipher. This may allow a man-in-the-middle attacker to downgrade the security of a TLS session to 512-bit export-grade cryptography, which is significantly weaker, allowing the attacker to more easily break the encryption and monitor or tamper with the encrypted stream.
Disclosure date: 2015-5-19

```
ssl-dh-params:
  VULNERABLE:
  Transport Layer Security (TLS) Protocol DHE_EXPORT Ciphers Downgrade MitM (Logjam)
    State: VULNERABLE
    IDs:  CVE:CVE-2015-4000  BID:74733
      The Transport Layer Security (TLS) protocol contains a flaw that is
      triggered when handling Diffie-Hellman key exchanges defined with
      the DHE_EXPORT cipher. This may allow a man-in-the-middle attacker
      to downgrade the security of a TLS session to 512-bit export-grade
      cryptography, which is significantly weaker, allowing the attacker
      to more easily break the encryption and monitor or tamper with
      the encrypted stream.
    Disclosure date: 2015-5-19
    Check results:
      EXPORT-GRADE DH GROUP 1
            Cipher Suite: TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
            Modulus Type: Safe prime
            Modulus Source: mod_ssl 2.2.x/512-bit MODP group with safe prime modulus
            Modulus Length: 512
            Generator Length: 8
            Public Key Length: 512
    References:
      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-4000
      https://weakdh.org
      https://www.securityfocus.com/bid/74733
```

2. Vulnerability

ssl-ccs-injection:
SSL/TLS MITM vulnerability (CCS Injection)
State: Vulnerable
Risk factor: **High**

OpenSSL before 0.9.8za, 1.0.0 before 1.0.0m, and 1.0.1 before 1.0.1h does not properly restrict processing of ChangeCipherSpec messages, which allows man-in-the-middle attackers to trigger use of a zero length master key in certain OpenSSL-to-OpenSSL communications, and consequently hijack sessions or obtain sensitive information, via crafted TLS handshake, aka the "CCS Injection" vulnerability.



3. Vulnerability

http-vuln-cve2011-3192:

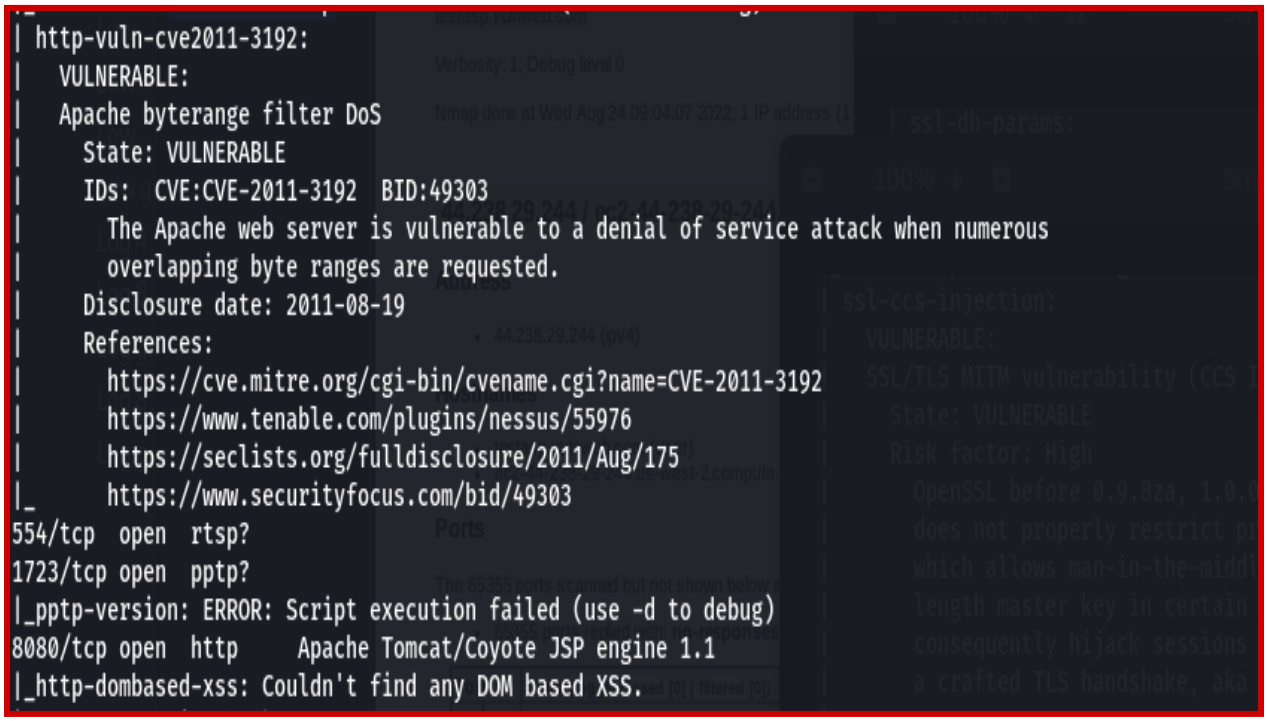
Apache byterange filter DoS

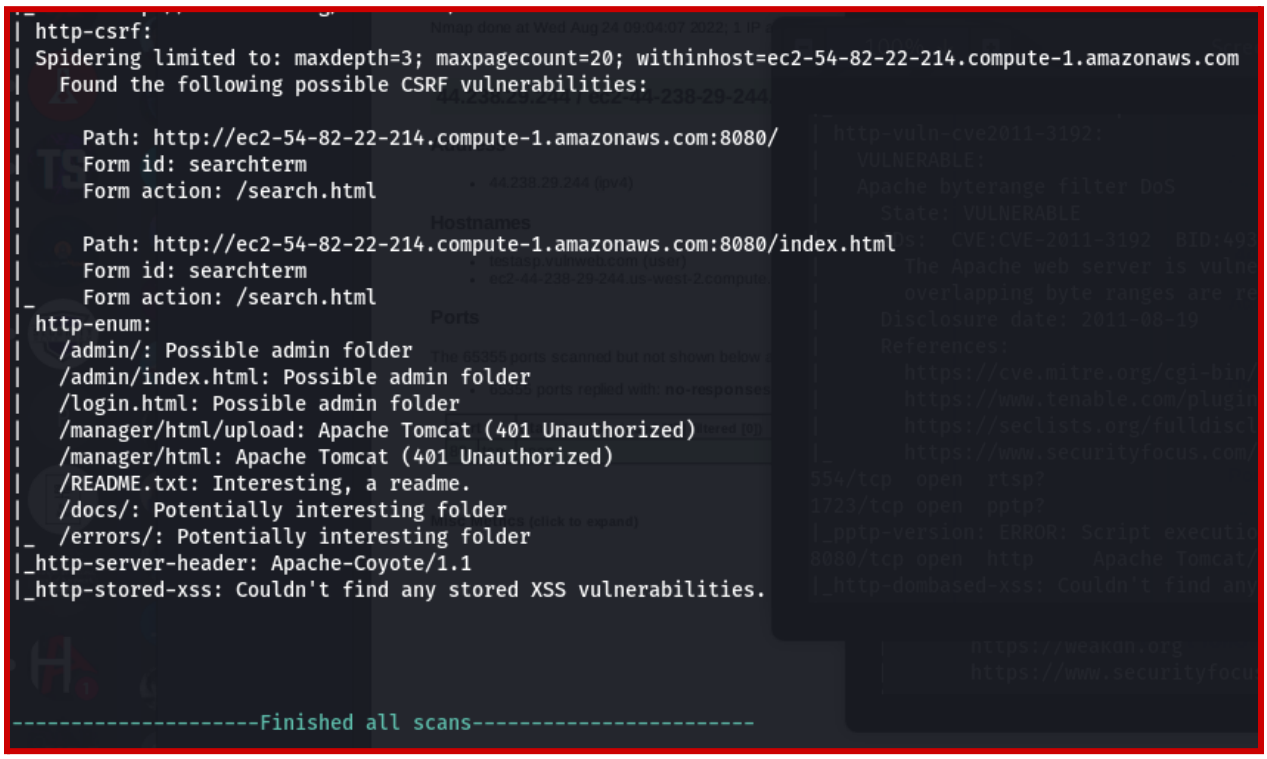
State: Vulnerable

IDs: CVE:CVE-2011-3192 BID:49303

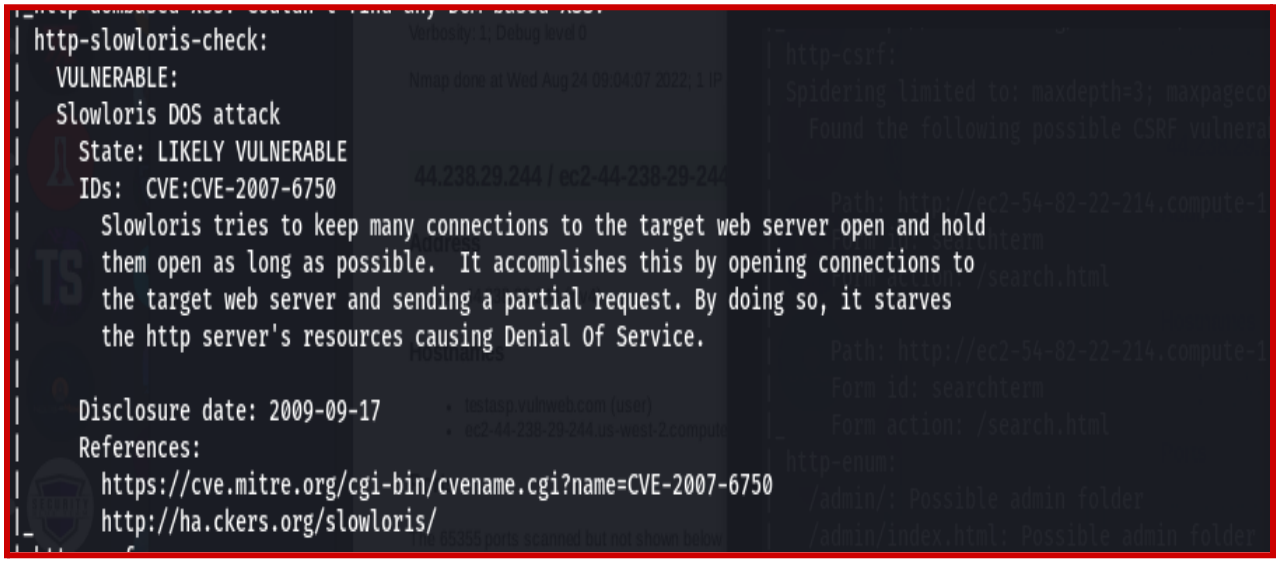
The Apache web server is vulnerable to a denial of service attack when numerous overlapping byte ranges are requested.

Disclosure date: 2011-08-19





4. Vulnerability



Nmap Scan Report - Scanned at Tue Aug 23 18:52:27 2022

Scan Summary | **zero.webappsecurity.com (54.82.22.214)**

Scan Summary

Nmap 7.40 was initiated at Tue Aug 23 18:52:27 2022 with these arguments:
nmap -v -oX=- --host-timeout=28800s -Pn -T4 -sT --webxml --max-retries=1 --open -p0-65355 zero.webappsecurity.com

Verbosity: 1; Debug level 0

Nmap done at Tue Aug 23 18:54:04 2022; 1 IP address (1 host up) scanned in 96.18 seconds

54.82.22.214 / ec2-54-82-22-214.compute-1.amazonaws.com / zero.webappsecurity.com

Address

- 54.82.22.214 (ipv4)

Hostnames

- zero.webappsecurity.com (user)
- ec2-54-82-22-214.compute-1.amazonaws.com (PTR)

Ports

The 65353 ports scanned but not shown below are in state: **filtered**

- 65353 ports replied with: **no-responses**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
80	tcp	open	http	syn-ack			
443	tcp	open	https	syn-ack			
8080	tcp	open	http-proxy	syn-ack			

High (CVSS: 7.5)
NVT: Apache HTTP Server < 2.4.48 NULL Pointer Dereference Vulnerability - Windows

Product detection result
cpe:/a:apache:http_server:2.2.6
Detected by Apache HTTP Server Detection Consolidation (OID: 1.3.6.1.4.1.25623.1
↪.0.117232)

Summary
Apache HTTP Server is prone to a NULL pointer dereference vulnerability.

Vulnerability Detection Result
Installed version: 2.2.6
Fixed version: 2.4.48
Installation
path / port: 443/tcp

Impact
Successful exploitation will allow an attacker to crash the server.

Solution:
Solution type: VendorFix
Update to version 2.4.48 or later.

Affected Software/OS
Apache HTTP Server before version 2.4.48 on Windows.

Vulnerability Insight
Apache HTTP Server protocol handler for the HTTP/2 protocol checks received request headers against the size limitations as configured for the server and used for the HTTP/1 protocol as well. On violation of these restrictions an HTTP response is sent to the client with a status code indicating why the request was rejected.
This rejection response was not fully initialised in the HTTP/2 protocol handler if the offending header was the very first one received or appeared in a footer. This led to a NULL pointer dereference on initialised memory, crashing reliably the child process.

Vulnerability Detection Method
Checks if a vulnerable version is present on the target host.
Details: Apache HTTP Server < 2.4.48 NULL Pointer Dereference Vulnerability - Windows
OID:1.3.6.1.4.1.25623.1.0.112904
Version used: 2021-08-24T09:01:06Z

Product Detection Result
Product: cpe:/a:apache:http_server:2.2.6
Method: Apache HTTP Server Detection Consolidation

High (CVSS: 10.0)
NVT: Apache HTTP Server End of Life (EOL) Detection (Windows)

Product detection result
cpe:/a:apache:http_server:2.2.6
Detected by Apache HTTP Server Detection Consolidation (OID: 1.3.6.1.4.1.25623.1
↪.0.117232)

... continues on next page ...

... continued from previous page ...

Summary
The Apache HTTP Server version on the remote host has reached the End of Life (EOL) and should not be used anymore.

Vulnerability Detection Result
The "Apache HTTP Server" version on the remote host has reached the end of life.
CPE: cpe:/a:apache:http_server:2.2.6
Installed version: 2.2.6
Location/URL: 443/tcp
EOL version: 2.2
EOL date: 2017-12-31

Impact
An EOL version of the Apache HTTP Server is not receiving any security updates from the vendor. Unfixed security vulnerabilities might be leveraged by an attacker to compromise the security of this host.

Site: http://zero.webappsecurity.com

Generated on Tue, 23 Aug 2022 18:53:02

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	4
Low	1
Informational	0
False Positives:	0

Alerts

Name	Risk Level	Number of Instances
Absence of Anti-CSRF Tokens	Medium	6
Cross-Domain Misconfiguration	Medium	16
Missing Anti-clickjacking Header	Medium	6
Vulnerable JS Library	Medium	2
X-Content-Type-Options Header Missing	Low	10

Alert Detail

Medium	Absence of Anti-CSRF Tokens
Description	No Anti-CSRF tokens were found in a HTML submission form. A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf. CSRF attacks are effective in a number of situations, including: * The victim has an active session on the target site. * The victim is authenticated via HTTP auth on the target site. * The victim is on the same local network as the target site. CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.
URL	http://zero.webappsecurity.com
Method	GET
Parameter	

Attack	
Evidence	<form action="/search.html" class="navbar-search pull-right" style="padding-right: 20px">
URL	http://zero.webappsecurity.com/
Method	GET
Parameter	
Attack	
Evidence	<form action="/search.html" class="navbar-search pull-right" style="padding-right: 20px">
URL	http://zero.webappsecurity.com/index.html
Method	GET
Parameter	
Attack	
Evidence	<form action="/search.html" class="navbar-search pull-right" style="padding-right: 20px">
URL	http://zero.webappsecurity.com/login.html
Method	GET
Parameter	
Attack	
Evidence	<form id="login_form" action="/signin.html" method="post" class="form-horizontal">
URL	http://zero.webappsecurity.com/online-banking.html
Method	GET
Parameter	
Attack	
Evidence	<form action="/search.html" class="navbar-search pull-right" style="padding-right: 20px">
URL	http://zero.webappsecurity.com/search.html?searchTerm=ZAP
Method	GET
Parameter	
Attack	
Evidence	<form action="/search.html" class="navbar-search pull-right" style="padding-right: 20px">
Instances	6
Solution	Phase: Architecture and Design Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid. For example, use anti-CSRF packages such as the OWASP CSRFGuard. Phase: Implementation Ensure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script. Phase: Architecture and Design Generate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330). Note that this can be bypassed using XSS. Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation.

Alerts

Name	Risk Level	Number of Instances
Absence of Anti-CSRF Tokens	Medium	6
Cross-Domain Misconfiguration	Medium	16
Missing Anti-clickjacking Header	Medium	6
Vulnerable JS Library	Medium	2
X-Content-Type-Options Header Missing	Low	10