

Forum:[Hack Forums](#)

Thread: [Help with Apache Web Server :\)](#)

Itsfr0sty - October 3, 2019, 1:28 pm

To start, I do have permission to do this from my engineering teacher. My objective is to change an image on the website to something of my own. The website only works on the school internet. To get to it an ip address is typed in the address bar. I'm not going to do anything malicious (he would know who done it), I just want to test my skills and further my knowledge. The Scan says there are vulnerabilities, but I just don't know where to start. I also included the open ports if that helps. Here are the results from an nikto scan:

- + Server: Apache/2.4.10 (Win32) OpenSSL/1.0.1i
- + Cookie MoodleSession created without the httponly flag
- + The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
- + Uncommon header 'content-style-type' found, with contents: text/css
- + Uncommon header 'content-script-type' found, with contents: text/javascript
- + The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
- + Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
- + OpenSSL/1.0.1i appears to be outdated (current is at least 1.1.1). OpenSSL 1.0.0o and 0.9.8zc are also current.
- + Web Server returns a valid response with junk HTTP methods, this may cause false positives.
- + DEBUG HTTP verb may show server debugging information. See
- + OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
- + /index.php?option=search&searchword=<script>alert(document.cookie);</script>: Mambo Site Server 4.0 build 10 is vulnerable to Cross Site Scripting (XSS).
- + OSVDB-2820: /index.php?dir=<script>alert('Vulnerable')</script>: Auto Directory Index 1.2.3 and prior are vulnerable to XSS attacks.
- + OSVDB-50552: /index.php?file=Liens&op=\"><script>alert('Vulnerable')</script>: Nuked-klan 1.3b is vulnerable to Cross Site Scripting (XSS).
- + /index.php?action=storenew&username=<script>alert('Vulnerable')</script>: SunShop is vulnerable to Cross Site Scripting (XSS) in the signup page. CA-200-02.

- + /index.php/ "><script><script>alert(document.cookie)</script><: eZ publish v3 and prior allow Cross Site Scripting (XSS).
- + OSVDB-50553:
/index.php/content/search/?SectionID=3&SearchText=<script>alert(document.cookie)</script>: eZ publish v3 and prior allow Cross Site Scripting (XSS).
- + OSVDB-50553:
/index.php/content/advancedsearch/?SearchText=<script>alert(document.cookie)</script>&PhraseSearchText=<script>alert(document.cookie)</script>&SearchContentClassID=-1&SearchSectionID=-1&SearchDate=-1&SearchButton=Search: eZ publish v3 and prior allow Cross Site Scripting (XSS).
- + OSVDB-38019: /?mod=<script>alert(document.cookie)</script>&op=browse: Sage 1.0b3 is vulnerable to Cross Site Scripting (XSS).
- + /config.php: PHP Config file may contain database IDs and passwords.
- + OSVDB-25497: /index.php?rep=<script>alert(document.cookie)</script>: GPhotos index.php rep Variable XSS.
- + OSVDB-12606: /index.php?err=3&email=\"><script>alert(document.cookie)</script>: MySQL Eventum is vulnerable to XSS in the email field.
- + OSVDB-2790: /index.php?vo=\"><script>alert(document.cookie);</script>: Ralusp Sympoll 1.5 is vulnerable to Cross Site Scripting (XSS).
- + OSVDB-3268: /Agent/: Directory indexing found.
- + OSVDB-3092: /Agent/: This might be interesting...
- + OSVDB-3092: /auth/: This might be interesting...
- + OSVDB-3268: /backup/: Directory indexing found.
- + OSVDB-3092: /backup/: This might be interesting...
- + OSVDB-3268: /install/: Directory indexing found.
- + OSVDB-3092: /install/: This might be interesting...
- + OSVDB-3092: /lib/: This might be interesting...
- + OSVDB-3092: /login/: This might be interesting...
- + OSVDB-3268: /pix/: Directory indexing found.
- + OSVDB-3092: /pix/: This might be interesting...
- + OSVDB-3092: /README.TXT: This might be interesting...
- + OSVDB-3092: /readme.txt: This might be interesting...
- + OSVDB-4265: /help.php?chapter=<script>alert('Vulnerable')</script>: Squirrel Mail 1.2.7 is

vulnerable to Cross Site Scripting (XSS).

- + OSVDB-3268: /repository/: Directory indexing found.
- + /repository/: CRX WebDAV upload
- + 8672 requests: 0 error(s) and 38 item(s) reported on remote host
- + End Time: 2019-10-03 13:58:52 (GMT-4) (266 seconds)

Open ports:

Discovered open port 139/tcp

Discovered open port 445/tcp

Discovered open port 3389/tcp

Discovered open port 3306/tcp

Discovered open port 80/tcp

Discovered open port 135/tcp

Discovered open port 443/tcp

Discovered open port 49153/tcp

Discovered open port 49152/tcp

Discovered open port 49154/tcp

compalabee - October 3, 2019, 1:49 pm

(Oct 3, 2019 01:28 PM)Itsfr0sty Wrote: [\[CITE\]](#) To start, I do have permission to do this from my engineering teacher. My objective is to change an image on the website to something of my own. The website only works on the school internet. To get to it an ip address is typed in the address bar. I'm not going to do anything malicious (he would know who done it), I just want to test my skills and further my knowledge.

Fastest way? Do an MITM attack. Use it to change all image files into something of your choice. This would typically work if your webapp does not have SSL (which is most probable because it's on a localnet).

I remember an Android app could automate this, called Dsploit, which only works on rooted Android phones: [\[APP\]\[2.3+\] dSploit - The most complete and | Android Development and Hacking](#)

Dsploit used to be available in the Google Play Store. The problem was that the developer improved on it again and again -- until it got too powerful. In the end it could hack WEP WiFi, swap Javascript on hosts, change image files, among others. Google kicked it out of the Play Store (rendering it unfunded), but it continues to circulate.

EDIT:

I just realized you had port 139 and 445 on. Have you tried using an exploit based around NSA's EternalBlue exploit? I think Metasploit has a module that allows you to leave a meterpreter inside the host computer. Pwn it from there, done deal.

Itsfr0sty - October 3, 2019, 1:53 pm

(Oct 3, 2019 01:49 PM)compalabee Wrote: [\[CITE\]](#) Fastest way? Do an MITM attack. Use it to change all image files into something of your choice. This would typically work if your webapp does not have SSL (which is most probable because it's on a localnet).

I remember an Android app could automate this, called Dsploit, which only works on rooted Android phones: [\[\] dSploit - The most complete and | Android Development and Hacking\[/url\]](#)

Dsploit used to be available in the Google Play Store. The problem was that the developer improved on it again and again -- until it got too powerful. In the end it could hack WEP WiFi, swap Javascript on hosts, change image files, among others. Google kicked it out of the Play Store (rendering it unfunded), but it continues to circulate.

Will look into this thank you! But I was looking for a way to gain access to the server. Like being able to permanently change a particular photo. Also, the server is not https secured if that helps. Any advice is appreciated :)

compalabee - October 3, 2019, 2:17 pm

(Oct 3, 2019 01:53 PM)Itsfr0sty Wrote: [\[CITE\]](#) Will look into this thank you! But I was looking for a way to gain access to the server. Like being able to permanently change a particular photo. Also, the server is not https secured if that helps. Any advice is appreciated :)

Well, to start, if it's not HTTPS then it doesn't have SSL/TLS so, I'm fully betting it's vulnerable to an man-in-the-middle attack. The only downside is that it doesn't exactly change the file within the server -- it just changes the transmitted packets' data towards the end user.

Again, try checking out if you can use Metasploit on the machine. You mentioned a few ports opened. Here is the step-by-step of that, in case you need a refresher (I'm assuming your attacker machine is Debian-based, like Kali Rolling or Backtrack):

1. Use Nmap to check out the services in the ports (change 123.123.123.123 to target IP address)

Code `nmap -sV -T3 -O 123.123.123.123`

Better yet, go aggressive:

Code `nmap -A 123.123.123.123`

For legend on the parameters, (you probably know this but still)

Code `man nmap`

2. Install Metasploit. If you're using Kali official repositories, use

Code `apt update`

`apt install metasploit-framework`

3. Once installed, do the initiation sequence for Metasploit (this is important)

Start PostgreSQL

Code `/etc/init.d/postgresql start`Use the automated Metasploit Database Builder

Code `msfdb init`Finally, start the Metasploit console

Code `msfconsole`

4. Search around in the console for a specific Metasploit Module to do the job

Code `search type:exploit <name of port service>`

Note: Judging by your nikto scan, your target might be using Windows. I have very little experience bypassing active IDS in Windows, so I can't help you there.

kayla_90 - *October 4, 2019, 9:16 pm*

have you tried to upload an image thru the cross site vulnerability?