ITP-325 FINAL PROJECT

Penetration Testing Report | April 28, 2023

Group #2:
Matt Chen
Mantej Lamba
Caroline Finnerty
Jack Hensel
Lucas Halberg

Table of Contents

Executive Summary

Group 2 was hired by Professor Stefan McGregor and Client Point of Contact Patrick Rabin to run a penetration test across three specified networks on Practical Pentest Labs to find its vulnerabilities and any unprotected entry points into the network system. As white-hat hackers, we simulated an attack on the network with the objective of exploiting vulnerabilities and escalating privileges to root, or administrator, access. To find these vulnerabilities, we first had to run enumeration scans on the range of IP addresses within the specified networks.

In sum, we found 238 vulnerabilities across 7 different hosts, including 29 medium-to-low level vulnerabilities and 207 pieces of information that could not be assigned a severity rating. The last two vulnerabilities found included one deemed critical risk and one deemed high risk both on the 10.0.3.11 host. These will likely be our easiest entry points into the network and thus a key focus of the engagement. They are as follows:

1. **Critical**: NFS Exported Share Information Disclosure
2. **High**: NFS Shares World Readable

Our team was able to successfully exploit the critical vulnerability of NFS Exported Share Information Disclosure. This implies that a malicious attacker may carry out identical actions, endangering the functionality and security of the entire network. Further detail regarding our precise approach to exploitation as well as recommendations and other essential information will be provided later in this report.

Scope of Services

For this engagement, we were asked to pose as external unauthorized intruders to assess the security processes & controls of the Pentest Training lab environment offered by Practical Pentest Labs. We did this in an ethical manner that did not disrupt the critical operations of Pentest labs or compromise the security of anyone on our engagement team following the Rules of Engagement set by the client. This was a purely remote penetration test engagement. The initial vulnerability assessment consisted of scans of the applications, servers, and workstations contained within the virtual lab infrastructure. All penetration and exploitation was conducted from Kali Linux virtual machines connected to the Pentest labs environment via a VPN.

We were limited both to non-destructive testing techniques (no files or data critical to the operations of the lab environment were modified or deleted) and to the systems identified in the Rules of Engagement, which included all IP addresses within the network ranges 10.0.1.0/24, 10.0.2.0/24 and 10.0.3.0/24. There were a total of 18 host machines available in the network,

including both Windows & Linux machines. We were not granted permission to scan or exploit the following networks: 10.0.20.0/24, 10.0.22.0/24, or any Student Kali systems.

All of the tools we are familiar with and planned to use were allowed. Any tools created or used by APTs, the dark web, or ransomware campaigns were reasonably off-limits. Lastly, simulated attack methods not allowed on this engagement beyond physical penetration testing included DoS attacks, disrupting traffic to or from the networks, or social engineering tactics.

Pentest Methodology

Pentest methodology is the process of testing a computer system or network for vulnerabilities in order to identify security risks and provide recommendations for mitigation. The first step of the methodology is vulnerability assessment, where we used tools such as Nessus and nmap to scan the provided networks of Pentest Practical Labs and identify potential vulnerabilities. We discuss these vulnerabilities in our first section titled, *Vulnerability Assessment*. The results are analyzed according to the Common Vulnerability Scoring System (CVSS) to determine the severity of each vulnerability. Following this, the findings are assessed in the *Detailed Vulnerability Findings* section, which includes vulnerability information, severity rating, affected systems and recommendations that can be implemented to fix each vulnerability. The *Penetration Test* assessment provides a walkthrough of the different steps and methods of the exploitation as well as *Recommendations* for fixing these vulnerabilities to prevent future exploitations.

Vulnerability Assessment

The first step was to run a Nessus vulnerability scan on networks 10.0.1.0/24, 10.0.2.0/24, and 10.0.3.0/24. Our team found a total of 31 vulnerabilities: one critical vulnerability and one high risk vulnerability, as well as 6 medium risk and 4 low risk vulnerabilities that repeat across some hosts. One column of information Nessus provides in the vulnerability scan results are plugins, which are programs created by Tenable Research to detect vulnerabilities. Nessus detected and ranked these vulnerabilities according to the industry standard, the Common Vulnerability Scoring System (CVSS). These vulnerability scores are derived from a calculation of the ease of exploitation along with the impact of the exploit. This metric allows companies to prioritize the responses and resources necessary to defend against malicious actors launching attacks as shown in the table on the next page. In our case, the highest scores pointed us to the easiest target and methods to start our penetration test phase with.

Severity	CVSS	Name	Plugin	Host(s)
Critical	10.0	NFS Exported Share Information Disclosure	11356	10.0.3.11
High	7.5	NFS Shares World Readable	42256	10.0.3.11
Medium	6.1	JQuery 1.2 < 3.5.0 Multiple XSS	136929	10.0.1.6

Severity	CVSS	Name	Plugin	Host(s)
Medium	5.3	Browsable Web Directories	40984	10.0.3.10 10.0.1.14 10.0.1.12 10.0.1.6
Medium	5.3	Web Server info.php / phpinfo.php Detection	11229	10.0.3.10
Medium	4.3	SSH Weak Algorithms Supported	90317	10.0.1.14 10.0.1.12 10.0.1.6 10.0.3.11
Medium	4.3	Web Application Potentially Vulnerable to Clickjacking	85582	10.0.1.14 10.0.1.12 10.0.1.6 10.0.1.4
Low	3.7	SSH Weak Key Exchange Algorithms Enabled	153953	10.0.1.6 10.0.1.12 10.0.1.14 10.0.3.11
Low	2.6	Web Server Transmits Cleartext Credentials	26194	10.0.1.4 10.0.1.6 10.0.1.14
Low	2.6	SSH Server CBC Mode Ciphers Enabled	70658	10.0.1.6 10.0.1.12 10.0.1.14 10.0.3.11
Low	2.6	SSH Weak MAC Algorithms Enabled	71049	10.0.1.6 10.0.1.12 10.0.1.14 10.0.3.11

Detailed Vulnerability Findings

NFS Exported Share Information Disclosure

Description: This vulnerability allows a user's file system to be mounted from a remote location. This allows a malicious actor to access sensitive information or execute remote code on the target machine.

Severity Rating: Critical

Affected System: 10.0.3.11

Recommendation: The network administrators should ensure that the access control list (ACL) only provides access to approved users or IP addresses. Using firewalls can also limit unauthorized access to the NFS service. Finally, install the most recent software patches and upgrades to the NFS server and its clients to fix any known vulnerabilities. You may also consider encrypting NFS traffic between servers and clients using virtual private networks (VPNs).

Source: https://www.tenable.com/plugins/nessus/11356

NFS Shares World Readable

Description: Since the NFS shares are publicly accessible, anyone can view their data without logging in. This flaw enables unauthorized access to private data, which could result in data loss, data manipulation, or other security lapses. The danger of exploitation is reduced by the requirement that an attacker have network access in order to exploit the vulnerability.

Severity Rating: High

Affected System: 10.0.3.11

Recommendation: The shares should be configured by the NFS server administrator to limit access to only authorized users. Access restrictions can be put in place on the NFS server to limit access to just approved clients or file permissions can be changed to prevent world-readable access. To prevent unwanted access to sensitive data, the NFS server should also be set to demand authentication and encryption. An example of encryption would be to implement a firewall.

Source: https://www.tenable.com/plugins/nessus/42256

JQuery 1.2 < 3.5.0 Multiple XSS

Description: Before version 3.5.0, there were numerous Cross-Site Scripting (XSS) vulnerabilities in the jQuery library. The jQuery function's poor sanitization of user-provided input is the cause of the vulnerabilities.

Severity Rating: Medium

Affected Systems: 10.0.1.6

Recommendation: To fix these flaws, the web application developer should upgrade to the most recent version of the jQuery library (version 3.5.0 or higher). Additionally, users should be made aware of the dangers of accessing unknown websites and clicking on unidentified links.

Source: https://www.tenable.com/plugins/nessus/136929

Browsable Web Directories

Description: We found a number of browsable web directories from the Affected Systems. This is a security concern as sensitive information might be located in those browsable directories. An attacker could discover and use this information to leverage against the host.

Severity Rating: Medium

Affected Systems: 10.0.3.10, 10.0.1.14, 10.0.1.12, 10.0.1.6

The following web directories were exploited by connecting to the Practical Pentest Labs OpenVPN, then using an internet search engine to look up the following URLs:

- http://10.0.1.6/images/
- http://10.0.1.6/images/design/
- http://10.0.1.6/images/eshop/
- http://10.0.1.6/images/portret/
- http://10.0.1.6/inc/
- http://10.0.1.6/inc/contacts/
- http://10.0.1.6/js/
- http://10.0.1.6/tools/
- http://10.0.1.6/user/
- http://10.0.3.10/
- http://10.0.1.12/admin/
- http://10.0.1.12/admin/uploads/
- http://10.0.1.14/

Recommendation: To remediate this issue, we recommend disabling browsable web directories at the website level, so they cannot be accessed by people without proper credentials.

Source: https://www.tenable.com/plugins/nessus/40984

Web Server info.php / phpinfo.php Detection

Description: The remote web server contains a PHP script that is prone to an information disclosure attack. The script will print out information on both the host and target machine.

An attacker can obtain information such as:
- Exact PHP version.
- Exact OS and its version.
- Details of the PHP configuration.
- Internal IP addresses.
- Server environment variables.
- Loaded PHP extensions and their configurations.

Severity Rating: Medium

Affected Systems: 10.0.3.10

Recommendation: To prevent this from happening we suggestto remove pages that call php from the web server and disable php by using global php configurations.

Source: https://www.tenable.com/plugins/nessus/11229

SSH Weak Algorithms Supported

Description: We detected that the remote SSH server is configured to use the Arcfour stream cipher or no cipher at all. RFC 4253 advises against using Arcfour due to an issue with weak keys.

Severity Rating: Medium

Affected Systems: 10.0.1.14, 10.0.1.12, 10.0.1.6, 10.0.3.11

Recommendation: We suggest that you contact your vendor to receive an updated and secure version of software to produce ciphers.

Source: https://www.tenable.com/plugins/nessus/90317

Web Application Potentially Vulnerable to Clickjacking

Description: In the remote web server, the X-Frame-Options response header or the Content-Security-Policy 'frame-ancestors' response header is not set in all content responses. This may

leave the site vulnerable to a clickjacking or UI redress attack, where an attacker may deceive a user into clicking on a section of the vulnerable page that differs from the user's perceived normal view of the page. As a consequence, the user could unwittingly execute fraudulent or malicious transactions.

Severity Rating: Medium

Affected Systems: 10.0.1.14, 10.0.1.12, 10.0.1.6, 10.0.1.4

The following pages do not use a clickjacking mitigation response header and contain a clickable event:

- http://10.0.1.4/
- http://10.0.1.6/
- http://10.0.1.6/inc/admin2_box.php
- http://10.0.1.6/inc/eshop.php
- http://10.0.1.6/inc/eshop_rekapitulace.php
- http://10.0.1.6/inc/help.php
- http://10.0.1.6/inc/help_box.php
- http://10.0.1.6/inc/kontakt.php
- http://10.0.1.6/inc/message.php
- http://10.0.1.6/inc/nastaveni.php
- http://10.0.1.6/inc/passchange.php
- http://10.0.1.6/inc/resend.php
- http://10.0.1.6/inc/search.php
- http://10.0.1.6/inc/slozka.php
- http://10.0.1.6/inc/soutez.php
- http://10.0.1.6/inc/soutez_best.php
- http://10.0.1.6/inc/soutez_box.php
- http://10.0.1.6/user/fpass.php
- http://10.0.1.6/user/fpass_box.php
- http://10.0.1.6/user/register.php
- http://10.0.1.6/user/register_main.php
- http://10.0.1.12/admin/upload.php
- http://10.0.1.14/register.php

Recommendation: To remediate this issue, add the X-Frame-Options or Content-Security-Policy HTTP header (including the 'frame-ancestors' directive) to the page's response. By doing so, it will prevent the page's content from being displayed on another website using the frame or iframe HTML tags.

Source: https://www.tenable.com/plugins/nessus/85582

SSH Weak Key Exchange Algorithms Enabled

Description: The remote SSH server is configured to allow key exchange algorithms which are considered weak.Using a SSH, two previously unknown parties to generate a shared key in plain sight, and have that secret remain private to the client and server.

Severity Rating: Low

Affected Systems: 10.0.1.6, 10.0.1.12, 10.0.1.14, 10.0.3.11

Recommendation: We suggest that you contact your vendor to receive an updated and secure version of the software.

Source: https://www.tenable.com/plugins/nessus/153953

Web Server Transmits Cleartext Credentials

Description: The remote web server contains several HTML form fields containing an input of type 'password' which transmit their information to a remote web server in cleartext.

Severity Rating: Low

Affected Systems: 10.0.1.4, 10.0.1.6, 10.0.1.14

Recommendation: To insure this issue does not occur make sure that every sensitive form transmits content over HTTPS or another secure server.

Source: https://www.tenable.com/plugins/nessus/26194

SSH Server CBC Mode Ciphers Enabled

Description: The SSH server is utilizing Cipher Block Chaining for encryption. As a result, the server may be susceptible to attacks that could recover the plaintext message from the ciphertext. Note that this plugin examines only the SSH server settings and not the software versions that may be vulnerable.

Severity Rating: Low

Affected Systems: 10.0.1.6, 10.0.1.12, 10.0.1.14, 10.0.3.11

Recommendation: Please contact the vendor or consult the product documentation to disable the use of CBC mode cipher encryption and enable the CTR or GCM cipher mode encryption as a suitable alternative.

Source: https://www.tenable.com/plugins/nessus/70658

SSH Weak MAC Algorithms Enabled

Description: The configuration of the remote SSH server allows for weak MD5 and 96-bit MAC algorithms. This poses a security risk as these algorithms are vulnerable to attack. Note that this plugin only checks for the options of the SSH server and does not check for vulnerable software versions.

Severity Rating: Low

Affected Systems: 10.0.1.6, 10.0.1.12, 10.0.1.14, 10.0.3.11

Recommendation: It is recommended to disable these weak algorithms and use stronger ones. To solve this issue, contact the vendor or consult product documentation in order to disable MD5 and 96-bit MAC algorithms.

Source: https://www.tenable.com/plugins/nessus/71049

Penetration Test

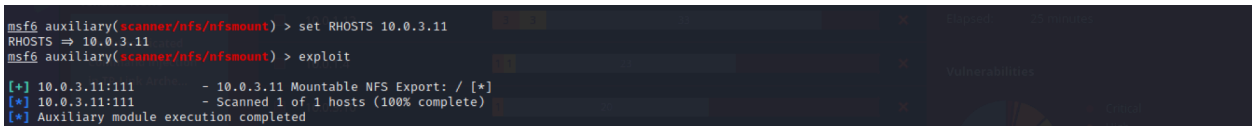
NFS Exported Share Information Disclosure

From filtering the Nessus scans we found a Metasploitable exploit for the critical **NFS Exported Share Information Disclosure** on the 10.0.3.11 host.

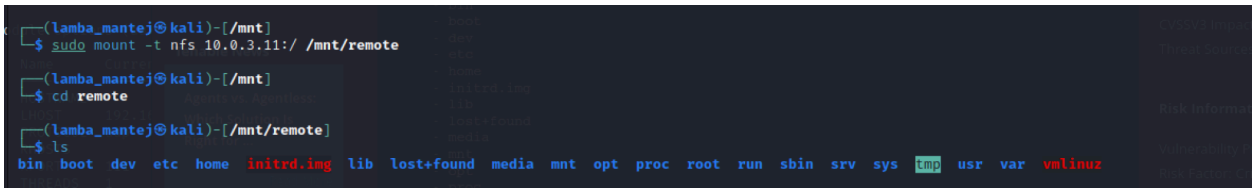
- To exploit this vulnerability, we decided to try the auxiliary/scanner/nfs/nfsmount metasploit module.
- From our research, we saw that this module is a tool used for scanning and exploiting vulnerabilities in NFS (Network File System) mounts.
    - NFS is a protocol used for sharing files and directories over a network, commonly used in UNIX and Linux environments. The nfsmount module can be used to scan a network for NFS shares and mount them as a local file system. This can be useful for accessing sensitive information on the network or for executing remote code on the target machine.

We were successfully able to exploit the vulnerability by using the metasploit module, as shown by the screenshots below. We set the RHOST to the IP address that the vulnerability came from,

and simply exploited it. It was successful since we got the message “Auxiliary module execution completed”.



After that, we went through the process of testing whether the NFS share is writable after mounting it locally using the mount command. We created a new directory on the local machine for the remote volume to be mounted in, and then used the mount command to attempt to link the remote volume to the local folder. We were not able to successfully mount the volume. If another attacker was, they could test whether it is writable by writing a file to the mounted directory using the echo command and then checking if the file exists on the remote end by reading it using the cat command.



You would end this test by using the umount command, since this unmounts the NFS share from the local directory. This should give us a file that contains valuable information, like hardcoded credentials, a passwords.txt, or an id_rsa.

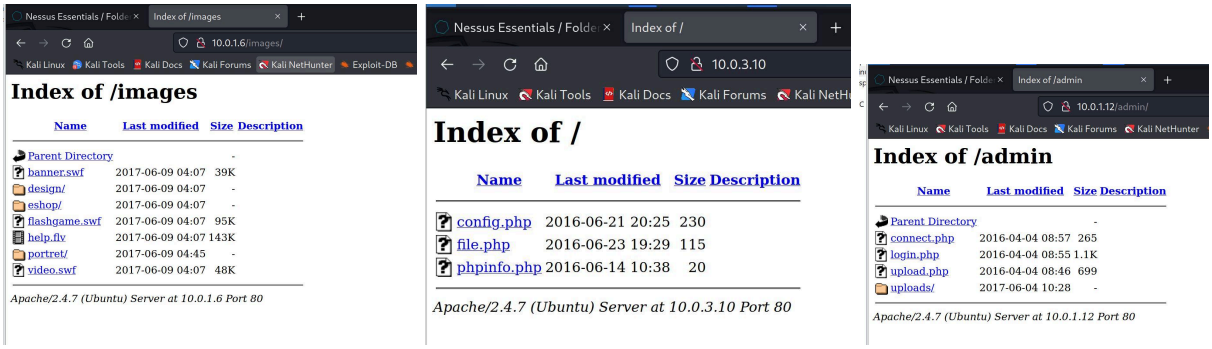
To prevent exploitation of this vulnerability, it is recommended to:

- **Review and modify NFS configuration**: Review the NFS server configuration and ensure that the access control list (ACL) is set to only allow access to authorized users or IP addresses;
- **Implement firewalls**: Use firewalls to restrict access to the NFS service to only authorized systems;
- **Patch or update software**: Ensure that the NFS server and clients are running the latest software updates and patches to address any known vulnerabilities;
- **Use VPNs**: Consider using virtual private networks (VPNs) to encrypt NFS traffic between servers and clients.

Browsable Web Directories

Our vulnerability scan identified a Browsable Web Directories vulnerability on the network. This vulnerability could allow unauthorized individuals to access the contents of web directories and potentially expose sensitive information. Web directories are folders or directories on a web server that contain website content such as web pages, images, videos, and other files. To exploit this vulnerability, we connected to the Practical Pentest Labs OpenVPN and used an internet search engine to look up URLs that led to web directories.

Using this method, we were able to access several web directories shown below:



To prevent further exploitation of this vulnerability, it is recommended that the client take the following steps:

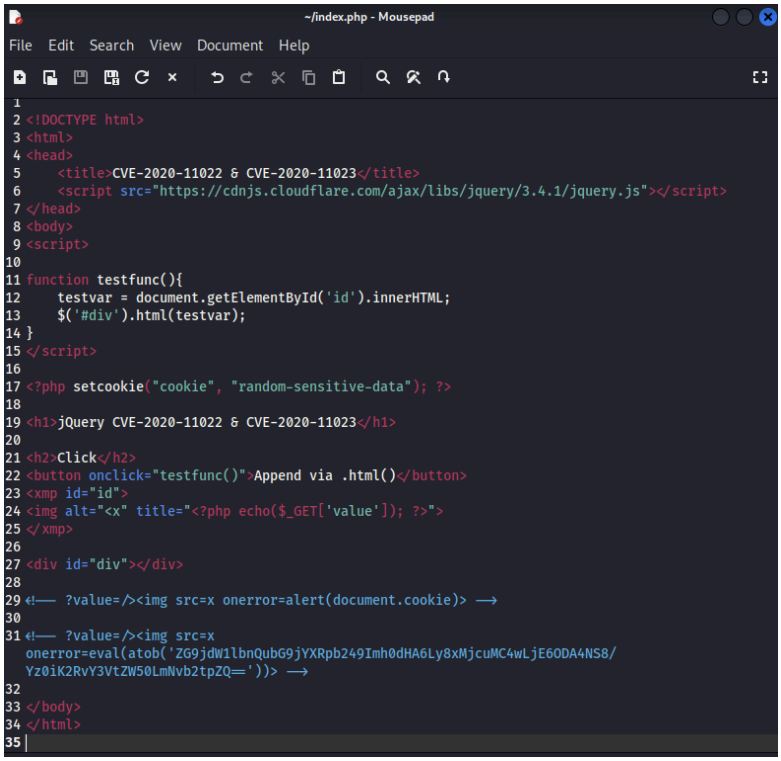
- **Review and modify directory permissions**: The company should review and modify directory permissions to ensure that only authorized users have access to sensitive directories.
- **Implement access control**: The company should implement access control measures to restrict access to sensitive directories and files.
- **Regularly scan for vulnerabilities**: The company should conduct regular vulnerability scans to identify potential vulnerabilities and take action to address them before they can be exploited.
- **Keep software up to date**: The company should ensure that all software and applications are up to date and that all security patches are installed in a timely manner to address any known vulnerabilities.

By following these recommendations, the client can better protect itself from potential attacks and minimize the risk of data breaches and other security incidents.

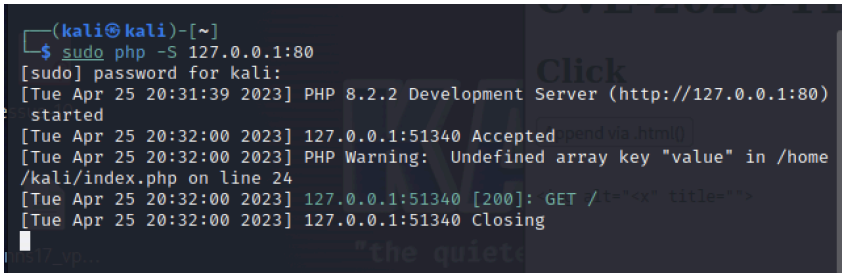
JQuery 1.2 < 3.5.0 Multiple XSS

This is a cross-site scripting (XSS) attack, which requires user interaction from the other side, such as clicking a malicious link. This was outside the scope of our engagement, but the following demonstrates how this attack would look if a user on the 10.0.1.6 host fell victim to it through a phishing attack for example. There are many ways to exploit this vulnerability. In this case, we showed how to steal the user's cookie.

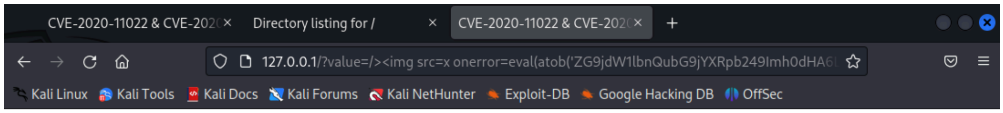
First, I saved a php script to my Kali Linux environment:



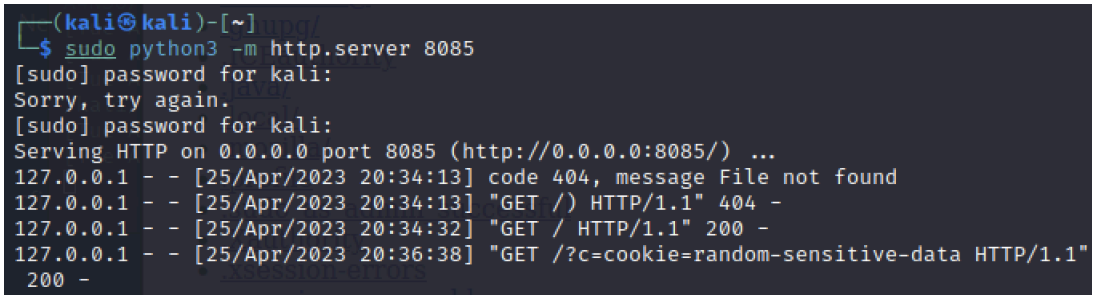
I then served a php web server on port 80. This port is where the XSS vulnerability lays as it is responsible for http, which handles unencrypted web traffic.



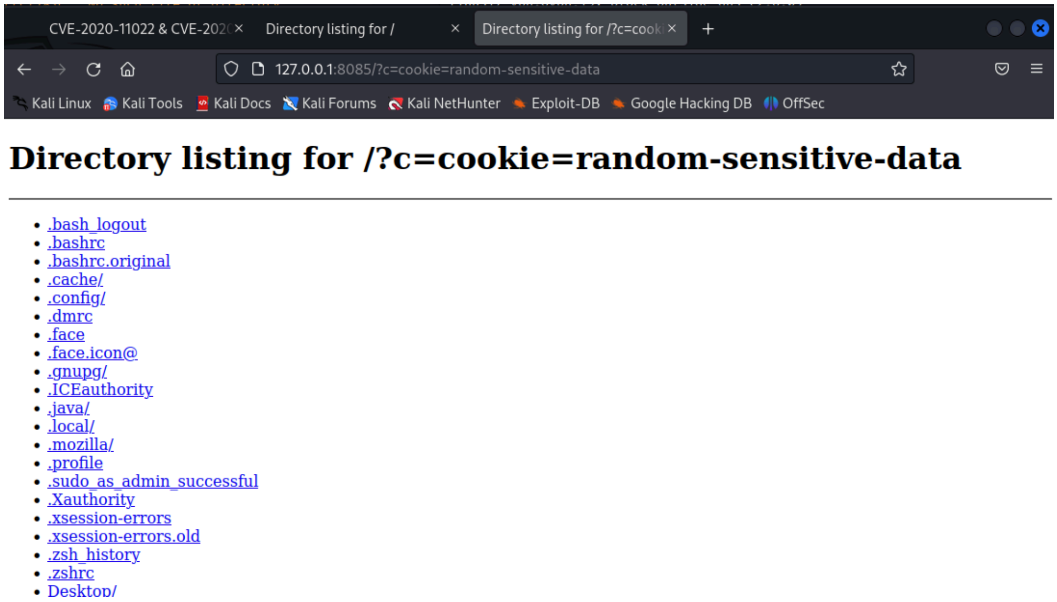
If the user were to open the link to my php webserver in a phishing attack or otherwise, they would see this html page created by the above php script:



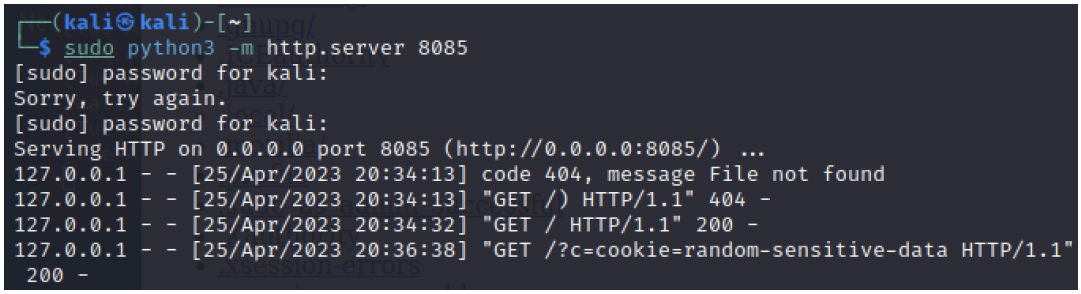
I then ran a python web server on port 8085. This port runs TCP, which establishes a connection between two hosts (the attacker and the victim).



When the user clicks the “append via .html()” button, the php file executes in their browser, displaying the cookie name. In this example, it is “random-sensitive-data”.



Since it executes in the browser, the attacker can now see the cookie name in their python web server logs. From there, the attacker can use for the cookie to impersonate the victim for myriad purposes. This is helpful for an attacker in situations where the cookie is for a session where the victim has already authenticated into something like their email server, so the attacker can access potentially sensitive information there.



To prevent exploitation of this vulnerability, it is recommended to:

- **Update software**: This vulnerability is patched in versions of jQuery 3.5.0 and later.
- **Educate users**: Anyone with access to these systems should be educated on how to recognize, avoid, and report suspicious links and phishing attempts;
- **Use HTTP and VPNs**: Consider enforcing all web traffic to be done via https and/or using virtual private networks (VPNs) to encrypt traffic between servers and clients.

Note: Multi-Factor Authentication (MFA) alone will not protect systems from this vulnerability when it is exploited to steal cookies.

Web Application Potentially Vulnerable to Clickjacking

A clickjacking attack is outside the scope of this engagement, but the following outlines how an attacker may exploit this vulnerability.
In order to create and use a clickjacking exploit on the remote web servers that do not set an X-Frame-Options response header or a Content-Security-Policy response header in all content responses, we would first identify the vulnerable web application using a vulnerability scanner, such as Nessus, or by manually inspecting the HTTP response headers.

Once we have identified the target application, we would create a malicious web page with an iframe or frame tag that would overlay the target website's content. The iframe or frame tag would be positioned so that it would be invisible to the user, and the user would be tricked into clicking on a different area of the vulnerable page than what they perceive it to be, leading to the stealing of credentials or execution of fraudulent transactions.

- To create a malicious web page with an iframe or frame tag:
    - Open a text editor and create a new file.
    - Add the HTML code for the malicious page. This code should include the iframe or frame tag that will overlay the target website's content.
    - Save the file with an appropriate name and the .html extension
    - Once you have created the malicious web page, you can host the page on your web server.
- You can use a web server software like Python's SimpleHTTPServer module to host the web page.

- For example, if using Python's SimpleHTTPServer module:
    - Open a command prompt or terminal on your computer.
    - Navigate to the directory where the malicious web page files are located using the cd command.
    - Start the web server by running the command python -m SimpleHTTPServer.
    - Verify that the web server is running by opening a web browser and navigating to http://localhost:8000 (if default port 8000 was used). The contents of the directory should be displayed in the browser.

Next we would get a target user to interact with the malicious web page by launching a phishing attack. This can be done through various means, such as sending a phishing email or enticing the target user to visit the malicious web page through social engineering tactics.

- To send a phishing email, you could craft an email that appears to be legitimate and trustworthy, such as pretending to be a well-known brand or authority figure. The email would include a link to the malicious web page, enticing the user to click on it. You may also include a call to action that urges the user to enter their login credentials or other sensitive information.
- You could also use social engineering tactics such as creating a fake login page that appears to be the legitimate one, and tricking the user into entering their credentials. This can be done by sending a link to the user or redirecting them to the malicious web page through other methods, such as pop-up ads or phishing kits.

To prevent exploitation of this vulnerability, it is recommended that the website owner returns the X-Frame-Options or Content-Security-Policy HTTP header with the page's response. This would prevent the page's content from being rendered by another site when using the frame or iframe HTML tags. The X-Frame-Options header specifies whether the browser should allow the page to be rendered in a frame or iframe. If this header is set to 'deny', the browser will not render the page in a frame or iframe at all. To do this:

- Identify the web server and locate the configuration file that controls the response headers.
- Open the configuration file and add the following line to set the X-Frame-Options header to 'deny': “ X-Frame-Options: DENY”

The Content-Security-Policy header with the 'frame-ancestors' directive specifies which domains are allowed to embed the page in a frame or iframe. By setting this directive to 'none', the browser will not allow any domains to embed the page.

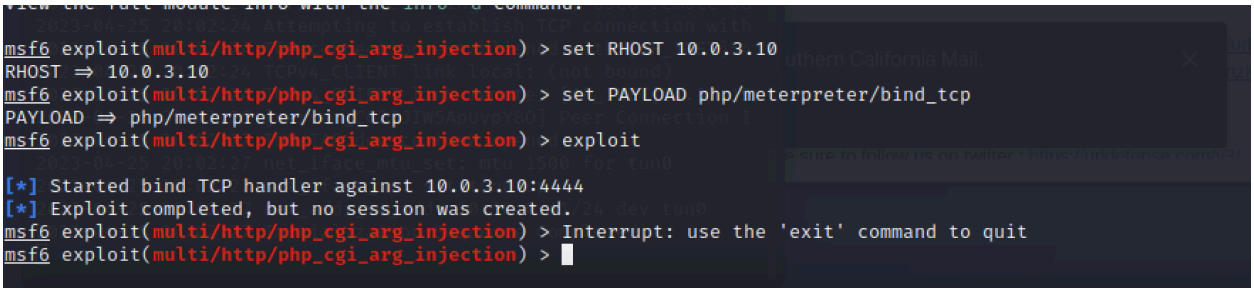
- If using Content-Security-Policy header, add the following line to set the frame-ancestors directive to 'none': “Content-Security-Policy: frame-ancestors 'none'”
- Save the changes to the configuration file and restart the web server for the changes to take effect.

Web Server info.php / phpinfo.php Detection

From our vulnerability scan, we saw that there was a Web server vulnerability, which references a page that displays information about the PHP configuration and other related modules of a web server. It is usually created for debugging or troubleshooting purposes, and it is often used by developers to ensure that the PHP environment is correctly configured. However, having this page publicly accessible on a web server can pose a security risk. The information displayed on this page can reveal the version of PHP, web server software, and other system information that could be useful to attackers looking to exploit vulnerabilities specific to the PHP and web server versions used by the server. In order to exploit this, we tried to use the exploit/multi/http/php_cgi_arg_injection module in Metasploit, which targets a vulnerability in PHP-CGI that allows remote code execution.

The vulnerability can be exploited by this module as the vulnerability occurs when PHP is used as a CGI script, and the script passes unvalidated user input. An attacker can inject a command into the variable, which will be executed by PHP. This vulnerability can be exploited to gain remote code execution on the target system.

We successfully ran this exploit on the 10.0.3.10 IP Address, but were not able to establish a session. We attempted to change our target system by setting the Target ID instead of the RHOST, but that still did not give way to system access. If we succeeded in this step, we would see a message from Meterpreter confirming we have a session on the target machine. From there, we could run a Ruby program that reads a list of IP addresses from a file, attempts to exploit a vulnerability in a PHP CGI program running on each IP address, and runs a payload on the target system.



Recommendations

Based on the vulnerabilities that we found and were able to exploit, the recommendations below detail best practices to secure the network and ensure its safety. Keep in mind that the list below is non-exhaustive but rather a priority list of action items that should be implemented as soon as possible.

1. **Disable the NFS service**: Disabling the NFS service on Linux systems is one technique to solve this problem. This will eliminate the possibility of information disclosure and stop attackers from gaining access to the exported shares.
2. **Protect the NFS shares**: If disabling the NFS service is not an option, the system owner should take precautions to protect the NFS share functionality. One way to accomplish this is to limit access to shares so that only users with proper permissions can read or write to them. Linux offers a "root squashing" ability which does not allow regular users to mount NFS as root.
3. **Patch the systems**: Keeping all software up to date is always highly recommended to patch known vulernabilities across the system. Specific recommendations follow each exploit such as updating jQuery to version 3.5.0 or later, but the most recent operating systems will enforce the most recent best practices.
4. **Implement access controls**: Using strong credentials across the entire organization will ensure the safety and security of sensitive information. Encouraging unique and lengthy usernames and passwords will add an extra layer of security, preventing the risk of breach. Multi-Factor Authentication is also advised to prevent attackers with stolen credentials from escalating privileges. Firewall implementation mitigates lateral and vertical escalation through the system. Access controls for specific services are outlined above.

Conclusion

In conclusion, Group 2, acting as white-hat hackers, conducted a thorough penetration test on the specified networks of Practical Pentest Labs with the aim of finding vulnerabilities and entry points that could be exploited. A total of 12 medium-to-low level vulnerabilities were found across 7 hosts and 207 pieces of information that could not be assigned a severity rating. The critical and high-risk vulnerabilities found on the 10.0.3.11 host were identified as the easiest entry points into the network and thus received significant attention during the engagement. Our team successfully exploited the critical vulnerability of NFS Exported Share Information Disclosure and the high-risk vulnerability of Browsable Web Directories, highlighting the potential risk posed to the network.

Contact Emails for Team Members

Matt Chen — mfchen@usc.edu
Caroline Finnerty — cafinner@usc.edu
Jack Hensel — jchensel@usc.edu
Lucas Halberg — lhalberg@usc.edu
Mantej Lamba — mlamba@usc.edu

Revision History and Quality Assurance (QA)

Date	QA Process	Status	Team Member
4/18	Report Writing Begins	Approved	Everyone, Team 2
4/20	Report Rough Draft Reviewed	Approved	Matt Chen
4/20	Report Rough Draft Reviewed	Approved	Caroline Finnerty
4/21	Report Rough Draft Finalized	Approved	Everyone, Team 2
4/25	Revising Rough Draft	Approved	Everyone, Team 2
4/28	Final Draft Reviewed	Approved	Everyone
4/28	Final Draft Reviewed	Approved	Everyone
4/28	Final Draft Submitted	Approved	Everyone