Journal of Social Research

P-ISSN: 2827-9832
E-ISSN: 2828-335x
http://ijsr.internationaljournallabs.com/index.php/ijsr

Evaluating the Effectiveness of Center of Internet Security Benchmark for Hardening Linux Servers Against Cyber Attacks

Bambang Irawan[1], Kholid Nur Sheha[2], Mosiur Rahaman[3], Nixon Erzed[4], Agus Herwanto[5]
Universitas Esa Unggul, Indonesia[1,2,4,5]
Asia University, Taiwan[3]
Email: bambang.irawan@esaunggul.ac.id, kholid.nursheha@student.esaunggul.ac.id, mosiurahaman@gmail.com, nixon@esaunggul.ac.id, agus.herwanto@esaunggul.ac.id

ABSTRACT

The security of operating systems is critical in safeguarding digital infrastructure, particularly server environments vulnerable to cyberattacks. One proven approach to enhancing OS security is hardening, which involves minimizing the system's attack surface. This study evaluates the effectiveness of the Center for Internet Security (CIS) Benchmark in hardening Ubuntu Server 22.04 against cyber threats. Using the PPDIOO framework, the research implemented hardening procedures via Ansible automation and conducted experimental tests comparing a hardened server against a standard (non-hardened) counterpart. Both servers were subjected to simulated attacks including DDoS, Port Scanning, Brute Force, Web Scanning, and Web Crawling. The results demonstrate a marked improvement in resistance for the hardened server, with attack success rates significantly reduced: 11% for DDoS (versus 94% on the standard server), 0% for Port Scanning, Brute Force, and Web Crawling (versus 20–100% on the standard server), and 67% for Web Scanning (versus 100% on the standard server). These findings underscore the substantial protective advantage conferred by the CIS Benchmark. The study contributes to the field by offering empirical evidence of CIS Benchmark's applicability to modern Linux environments and highlights the value of integrating automated hardening and attack simulations in cybersecurity practices. Future work should examine scalability across different OS platforms and real-world enterprise deployments.

Keywords: Operating System; OS Security; Hardening; CIS Benchmark; Ubuntu Server

INTRODUCTION

An Operating System (OS) is a collection of programs that play a crucial role in a computer system by controlling the execution of application programs and serving as an interface between applications and computer hardware. Additionally, the operating system manages and regulates user access to computer resources (Comer, 2025; Dieber et al., 2017; Irawan et al., 2024; Jaeger, 2022; Stallings, 2018). It implements access control to ensure that each user and application can only access the resources they are permitted to, preventing them from interfering with or accessing resources beyond their authorization. In this context, the operating system's security becomes critical to protect the computer system from threats and cyberattacks that could compromise data, privacy, and system operations. Successful attacks on an operating system not only have the potential to disrupt operations but can also lead to significant financial losses (Fitriani et al., 2023; Sari et al., 2024; Suhaemin & Muslih, 2021). For example, a ransomware attack affected Bank Syariah Indonesia in 2023, where customers could not access their mobile banking applications for several days. The hacker group claiming responsibility for the attack alleged that they had stolen 1.5 terabytes of data, including the personal information of customers and employees. They also threatened to sell this data on the dark web if their ransom demands were unmet (Angione et al., 2023; Ilonen et al., 2024; Leiritie, 2023; Sutanto et al., 2025).

Therefore, it is essential to understand and manage operating system security effectively. One common approach is to implement the hardening process on the operating system. Hardening reduces vulnerabilities in the operating system by removing unnecessary services, deactivating unused accounts, and adjusting security settings to comply with current industry standards. The CIS Benchmark is one of the security standards frequently used as a guide for the server hardening process, published by the Center for Internet Security in 2000. Its creation involved leading security experts and various organizations from industry, government, and academia. This guide is based on research and analysis of common vulnerabilities, typical attacks, and best practices in operating system security (Irfandi et al., 2022; Nieminen, 2025; Prastika et al., 2019; Vakhula et al., 2024).

Several similar studies have been conducted previously. The first study discussed using the CIS Benchmark guide as a reference for identifying and implementing best security practices on the Debian Server 8 Linux operating system; the audit results obtained a score of 70%. The second study analyzed server security using the Security Hardening process based on NIST Special Publication 800-123. The analysis results obtained 11 procedures that met the NIST SP 800-123 standard from 17 procedures recommended for implementation on the server. The third study focused on the implementation of hardening, UFW firewall, chmod, and chown to improve the security of the server operating system. The analysis results showed increased security after the hardening stages were carried out (Bachras, 2020; Deriyanto & Santoso, 2020; Ernawati et al., 2022; Tevault, 2020; Wijaya & Budiman, 2023). The difference between this and previous studies is that the CIS Benchmark method is applied to Ubuntu Server 22.04. In addition, this study involved testing attacks on the server to ensure the effectiveness of CIS Benchmark in improving operating system security (Hamidy & Yasin, 2024; Hyppönen, 2021).

This study evaluates the effectiveness of the Center for Internet Security (CIS) Benchmark in hardening Ubuntu Server 22.04 against cyber threats. This research presents a novel experimental evaluation of server hardening effectiveness using the CIS Benchmark applied specifically to Ubuntu Server 22.04, which is not addressed in prior studies. Previous works by Prastika et al. (2019) focused on Debian Server 8 using CIS auditing without direct attack testing; Irfandi et al. (2022) emphasized NIST SP 800-123 without quantitative benchmarking against real-world attacks; and Tevault (2023) explored a combination of UFW firewall, chmod, and chown techniques but did not evaluate standardized frameworks like CIS. In contrast, the current study not only applies CIS Benchmark through automation (Ansible Playbooks) but also directly simulates and measures resilience to five cyberattacks (DDoS, Port Scanning, Brute Force, Web Scanning, Web Crawling), demonstrating practical security gains. Thus, the research introduces a methodologically rigorous, implementation-based validation of CIS hardening through measurable threat resistance.

METHOD

This study follows the PPDIOO stages: Preparation, Planning, Design, Implementation, Operation, and Optimization.

1) Preparation Stage: The focus is identifying the existing problem through a literature review and defining the research problem, which is presented in the paper's introduction.
2) Planning Stage: This involves determining the research scenarios, including implementation, testing, and system requirements.
3) Implementation Scenario: The study uses an experimental approach comparing vulnerabilities on Ubuntu Server 22.04 before and after applying CIS Benchmark hardening guidelines. The independent variable is the hardening process, and the dependent variable is the server's vulnerability to attacks.
4) Testing Scenario: Vulnerability testing simulates cyberattacks such as DDoS, Port Scanning, Brute Force, Web Scanning, and Web Crawling, using common attacker techniques and tools.
5) System Requirements: Specifies the hardware and virtual environment setup, including admin and testing computers, Ubuntu servers (standard and hardened), all running on a KVM hypervisor, with detailed specifications provided in Table 1.

Table 1. Research Devices

No	Device	Spesifikasi	Software	Keterangan
1	Control	CPU 4 Core, RAM 8 GB, Disk 100 GB	Ubuntu Server 22.04 LTS, Ansible	Computer used for admin access and automation of hardening processes
2	Hardened Server	CPU 4 Core, RAM 8 GB, Disk 100 GB	Ubuntu Server 22.04 Lts	Ubuntu server that will undergo hardening with CIS Benchmark
3	Standard Server	CPU 4 Core, RAM 8 GB, Disk 100 GB	Ubuntu Server 22.04 Lts	Ubuntu server with standard configuration (non-hardened)
4	Attacker	CPU 4 Core, RAM 8 GB, Disk 100 GB	Kali Linux, Kali Linux Tools Packet	Computer used to test attacks on the Ubuntu server

The design stage focuses on creating the system topology, a crucial step in designing the network structure and data flow. In this stage, the topology design will be developed based on the research equipment described in the previous stage.

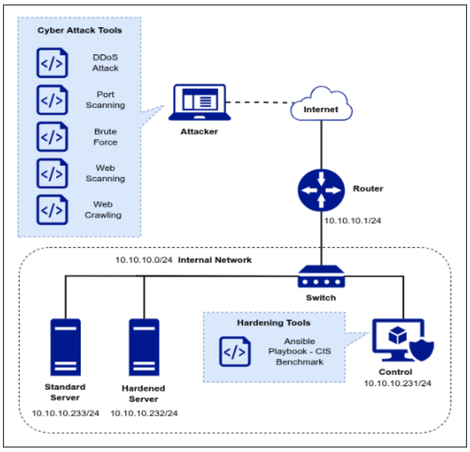


Figure 1. Three devices are connected to the internal network: Control, Hardened Server, and Standard Server. Meanwhile, the Attacker device is outside the server network and must be accessed through the internet.

The implementation phase focuses on the system's implementation according to the design from the previous phase. The research begins with creating a VM on KVM, installing the operating system, and installing the required packages according to the system requirements until the four devices are ready. The next step is to implement the selected server hardening method, in this case, using the CIS Benchmark guide. The hardening implementation begins by preparing the CIS Benchmark guide as an Ansible Playbook. Ansible allows researchers to define and implement hardening steps consistently and efficiently on Linux servers. When the Ansible playbook is run from the Control Server, the implementation will automatically start on the target server, namely the Hardened Server.

In the operation stage, comprehensive attack testing is conducted on the standard and hardened servers using the attack methods determined during the planning stage. The server attack testing is performed using the Test Bed method, conducted in an experimental environment to evaluate the performance, security, compatibility, and other aspects of the operating systems being tested. The attack testing will be carried out sequentially from the Attacker to both servers under test, ensuring a thorough evaluation of the system's security.

In the Optimization stage, the results obtained from the server testing process will be identified and analyzed. The results of this stage are presented in this paper's Results & Discussion section.

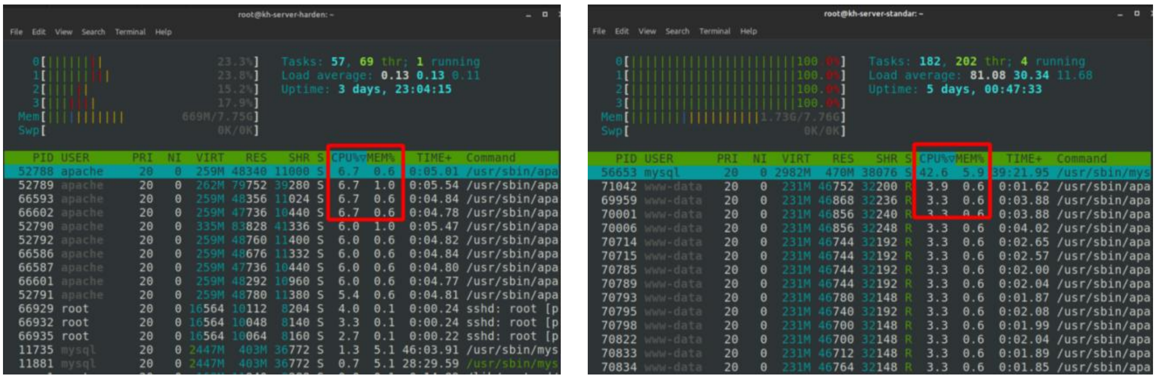
RESULTS AND DISCUSSION

This section outlines the attack testing results on the hardened and standard servers. This testing aims to evaluate and compare the security levels of both servers when facing similar attacks. By doing this, we can measure the effectiveness of the hardening steps applied based on CIS Benchmark recommendations in protecting the server from potential security threats and understand the security risks that remain in a server with default configurations.

DDoS Attack

A Distributed Denial of Service (DDoS) Attack is a type of cyberattack where numerous compromised computers or devices, often called bots or zombies, collaborate to render an online service inaccessible to users. This attack aims to hinder the availability of services or system resources by overwhelming the target with excessive network traffic.

In this research, DDoS attack testing was conducted to evaluate the resilience of services and resource capacity on hardened and standard servers. The attackers in this scenario specifically targeted the Web Server services at Layer 7 (the application layer of the OSI model), using a tool named L7kill. Figure 3(a) shows that resource usage on the hardened server remained consistently low throughout the DDoS attack test. In contrast, Figure 3(b) shows that the standard server experienced a significant increase in resource usage. The results of the DDoS attack test series are presented in Table 2. The test results show that the CPU usage on the hardened server is consistently much lower than on the standard server. The CPU usage on the hardened server ranges from 10.4% to 11.2%, indicating that this server has an adequate protection mechanism to handle DDoS attacks. In contrast, the standard server shows very high CPU usage, ranging from 92.8% to 94.7%, highlighting its vulnerability to DDoS attacks. These results indicate that the security enhancements and optimizations applied to the hardened server effectively mitigate the impact of DDoS attacks, which is critical to maintaining service availability and performance.


root@kh-server-harden: ~
File Edit View Search Terminal Help
root@kh-server-standar: ~
File Edit View Search Terminal Help

```
    0[||||||||                23.3%]   Tasks: 57, 69 thr; 1 running
    1[|||||||||               23.8%]   Load average: 0.13 0.13 0.11
    2[||||||                  15.2%]   Uptime: 3 days, 23:04:15
    3[|||||||                 17.9%]
  Mem[||||||||||||       669M/7.75G]
  Swp[                        0K/0K]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU%▽MEM%   TIME+  Command
52788 apache     20   0  259M 48340 11000 S  6.7  0.6  0:05.01 /usr/sbin/apa
52789 apache     20   0  262M 79752 39280 S  6.7  1.0  0:05.54 /usr/sbin/apa
66593 apache     20   0  259M 48356 11024 S  6.7  0.6  0:04.84 /usr/sbin/apa
66602 apache     20   0  259M 47736 10440 S  6.7  0.6  0:04.78 /usr/sbin/apa
52790 apache     20   0  335M 83828 41336 S  6.0  1.0  0:05.47 /usr/sbin/apa
52792 apache     20   0  259M 48760 11400 S  6.0  0.6  0:04.82 /usr/sbin/apa
66586 apache     20   0  259M 48676 11332 S  6.0  0.6  0:04.84 /usr/sbin/apa
66587 apache     20   0  259M 47736 10440 S  6.0  0.6  0:04.80 /usr/sbin/apa
66601 apache     20   0  259M 48292 10960 S  6.0  0.6  0:04.77 /usr/sbin/apa
52791 apache     20   0  259M 48780 11380 S  5.4  0.6  0:04.81 /usr/sbin/apa
66929 root       20   0 16564 10112  8204 S  4.0  0.1  0:00.24 sshd: root [p
66932 root       20   0 16564 10048  8140 S  3.3  0.1  0:00.24 sshd: root [p
66935 root       20   0 16564 10064  8160 S  2.7  0.1  0:00.22 sshd: root [p
11735 mysql      20   0 2447M  403M 36772 S  1.3  5.1 46:03.91 /usr/sbin/mys
11881 mysql      20   0 2447M  403M 36772 S  0.7  5.1 28:29.59 /usr/sbin/mys
```

(a)

```
    0[||||||||||||||||||||||||100.0%]   Tasks: 182, 202 thr; 4 running
    1[||||||||||||||||||||||||100.0%]   Load average: 81.08 30.34 11.68
    2[||||||||||||||||||||||||100.0%]   Uptime: 5 days, 00:47:33
    3[||||||||||||||||||||||||100.0%]
  Mem[|||||||||||||||||||1.73G/7.76G]
  Swp[                          0K/0K]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU%▽MEM%   TIME+  Command
56653 mysql      20   0 2982M  470M 38076 S 42.6  5.9 39:21.95 /usr/sbin/mys
71042 www-data   20   0  231M 46752 32200 R  3.9  0.6  0:01.62 /usr/sbin/apa
69959 www-data   20   0  231M 46868 32236 R  3.3  0.6  0:03.88 /usr/sbin/apa
70001 www-data   20   0  231M 46856 32240 R  3.3  0.6  0:03.88 /usr/sbin/apa
70006 www-data   20   0  231M 46856 32248 R  3.3  0.6  0:04.02 /usr/sbin/apa
70714 www-data   20   0  231M 46744 32192 R  3.3  0.6  0:02.65 /usr/sbin/apa
70715 www-data   20   0  231M 46744 32192 R  3.3  0.6  0:02.57 /usr/sbin/apa
70785 www-data   20   0  231M 46744 32192 R  3.3  0.6  0:02.00 /usr/sbin/apa
70789 www-data   20   0  231M 46744 32192 R  3.3  0.6  0:02.04 /usr/sbin/apa
70793 www-data   20   0  231M 46780 32148 R  3.3  0.6  0:01.87 /usr/sbin/apa
70795 www-data   20   0  231M 46740 32192 R  3.3  0.6  0:02.08 /usr/sbin/apa
70798 www-data   20   0  231M 46700 32148 R  3.3  0.6  0:01.99 /usr/sbin/apa
70822 www-data   20   0  231M 46700 32148 R  3.3  0.6  0:02.04 /usr/sbin/apa
70833 www-data   20   0  231M 46712 32148 R  3.3  0.6  0:01.89 /usr/sbin/apa
70834 www-data   20   0  231M 46764 32148 R  3.3  0.6  0:01.85 /usr/sbin/apa
```

(b)

Figure 2. Comparison of DDoS attack testing results on (a) the hardened server and (b) the standard server.

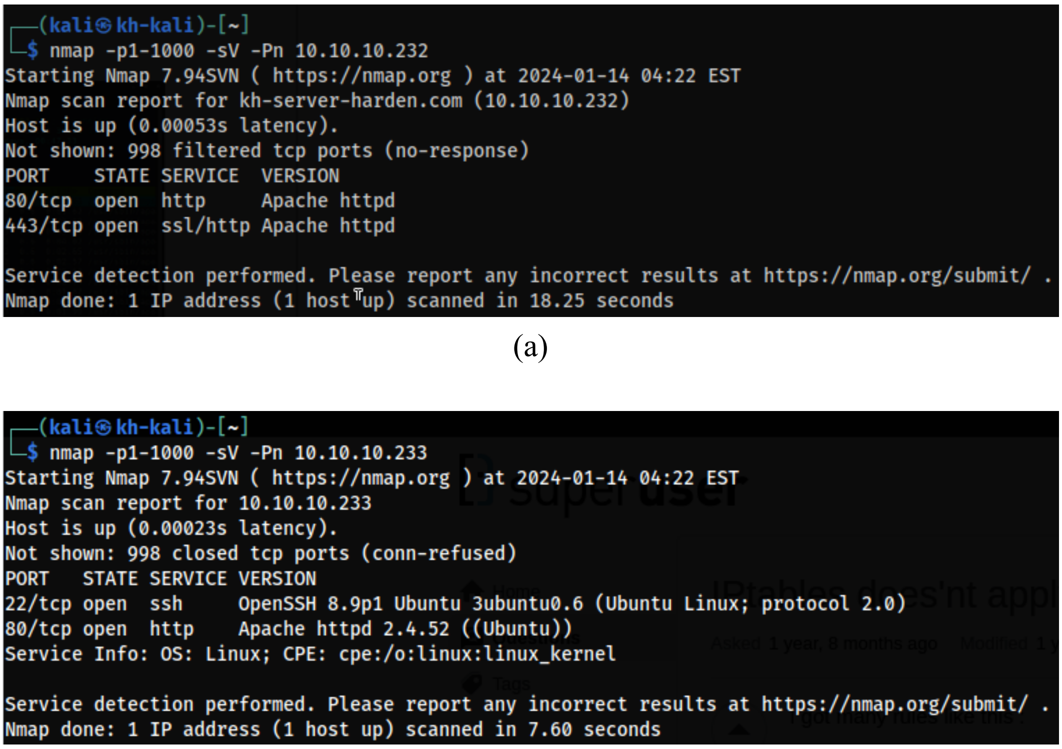
Table 2. The results of DDoS attack testing

No	DDoS Attack Testing (L7kill Tool)	Attack Duration	CPU Usage on Hardened Server	CPU Usage on Standard Server
1	Test 1	1 Minute	10,4 %	94,7 %
2	Test 2	2 Minutes	11,2 %	93,9 %
3	Test 3	3 Minutes	11,1 %	92,8 %
4	Test 4	4 Minutes	11,2 %	93,5 %
5	Test 5	5 Minutes	10,8 %	93,0 %

Port Scanning

Port scanning is a technique where an attacker attempts to discover open ports on the target system or computer network. These ports serve as entry or exit points that software uses to communicate over the network. Port scanning attacks aim to identify active ports vulnerable to further exploitation.

Port scanning tests are conducted in this phase using the Nmap tool, an industry standard for network exploration and security auditing. Figure 4(a) shows the results of a port scan on the harden server, where two ports are open: port 80 and port 443. In addition, 998 TCP ports are filtered and do not respond. This result indicates that a firewall or other security mechanism is hiding the status of these ports, thus not providing Nmap with enough information to determine whether the ports are open or closed. Figure 4(b) shows the results of a port scan on a standard server where two ports are open: port 22 and port 80. In addition, it is noted that the remaining 998 TCP ports are closed and refusing connections. This result indicates that these ports do not have any applications or services actively listening, and the server is explicitly refusing connections on these ports. The results of the port scanning attack series based on specific ports are presented in Table 3. The data shows that the hardened server consistently returns a status of "filtered" for the ports in every test, resulting in a 0% success rate for the attack. In contrast, the standard server has one port, 22/ssh, in the "open" state, and the other four ports show the "closed" state due to the absence of applications listening for requests from port scanning activity, resulting in a 20% success rate for the attack. These results indicate that the hardened server effectively blocks port scanning attempts compared to the standard server, which does not have additional firewall protection in its operating system.



```
┌──(kali㉿kh-kali)-[~]
└─$ nmap -p1-1000 -sV -Pn 10.10.10.232
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-14 04:22 EST
Nmap scan report for kh-server-harden.com (10.10.10.232)
Host is up (0.00053s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE  VERSION
80/tcp  open  http     Apache httpd
443/tcp open  ssl/http Apache httpd

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.25 seconds
```

(a)

```
┌──(kali㉿kh-kali)-[~]
└─$ nmap -p1-1000 -sV -Pn 10.10.10.233
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-14 04:22 EST
Nmap scan report for 10.10.10.233
Host is up (0.00023s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.52 ((Ubuntu))
Service Info: 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 7.60 seconds
```

(b)

Figure 3. Comparison of Port Scanning attack testing results on (a) the hardened server and (b) the standard server.

Table 3. The results of port scanning attack testing

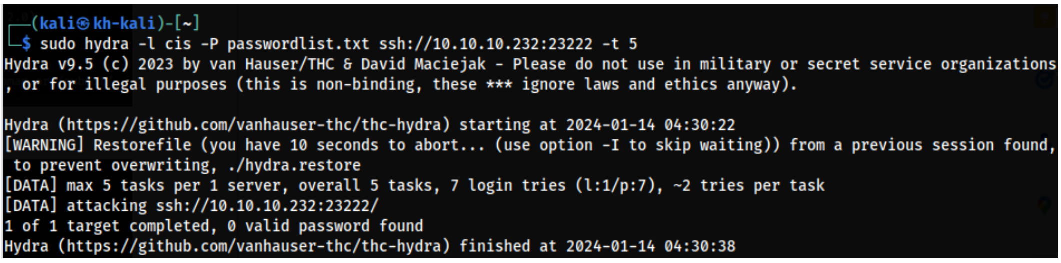
No	Port Scanning Attack Testing (Nmap)	Target Port	Hardened Server	Standard Server
1	Test 1	21 (FTP)	filtered	closed

No	Port Scanning Attack Testing (Nmap)	Target Port	Hardened Server	Standard Server
2	Test 2	22 (SSH)	filtered	open
3	Test 3	23 (TELNET)	filtered	closed
4	Test 4	25 (SMTP)	filtered	closed
5	Test 5	53 (DNS)	filtered	closed

Brute Force

Brute force is a cyberattack where an attacker attempts to gain unauthorized access to a server or system using the SSH (Secure Shell) protocol by repeatedly trying combinations of usernames and passwords using a specific wordlist. This attack relies on trial and error, where the Attacker automatically tries various password combinations until they find the correct one or gain access to the system.

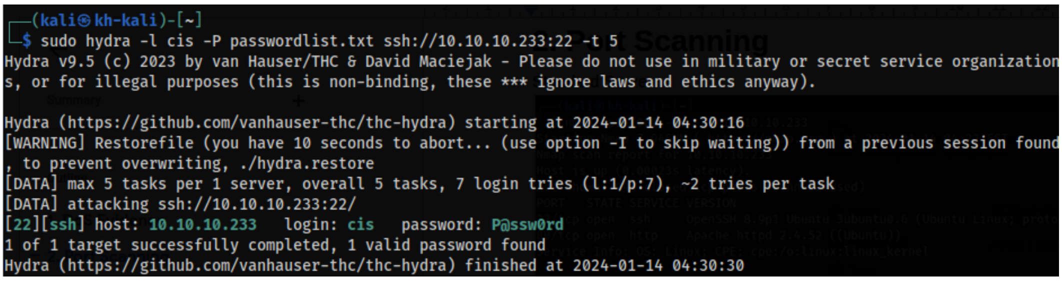
Brute Force attack testing is conducted to test the authentication system's resilience and security policies on the target servers, namely the hardened server and the standard server. In this scenario, the Attacker attempts a Brute Force attack using the Hydra tool, targeting the SSH service with common weak passwords (Dictionary Attack). Figure 5(a) shows the results of brute force attack tests on the harden server, which show that no passwords were successfully found by the Hydra tool. Figure 5(b) shows the results of brute force attack tests on the standard server, which show that one valid username and password combination has been successfully found. The Hydra tool indicates that the attack successfully broke the authentication on the server's SSH service using the given password list. The brute force attack test series is presented in Table 4. The data shows that the hardened server successfully repelled all attack attempts with a success rate of 0%. In contrast, every attack attempt on the standard server successfully found the server's credentials, resulting in a 100% attack success rate. These results indicate that the hardened server effectively repels brute force attacks compared to the standard server, which does not implement a complex password policy in the operating system.



```
┌──(kali㉿kh-kali)-[~]
└─$ sudo hydra -l cis -P passwordlist.txt ssh://10.10.10.232:23222 -t 5
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 2024-01-14 04:30:22
[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 5 tasks per 1 server, overall 5 tasks, 7 login tries (l:1/p:7), ~2 tries per task
[DATA] attacking ssh://10.10.10.232:23222/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-01-14 04:30:38
```

(a)



```
┌──(kali㉿kh-kali)-[~]
└─$ sudo hydra -l cis -P passwordlist.txt ssh://10.10.10.233:22 -t 5
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organization
s, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-01-14 04:30:16
[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 5 tasks per 1 server, overall 5 tasks, 7 login tries (l:1/p:7), ~2 tries per task
[DATA] attacking ssh://10.10.10.233:22/
[22][ssh] host: 10.10.10.233   login: cis   password: P@ssw0rd
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-01-14 04:30:30
```

(b)

Figure 4. Comparison of Brute Force attack testing results on (a) the hardened server and (b) the standard server .

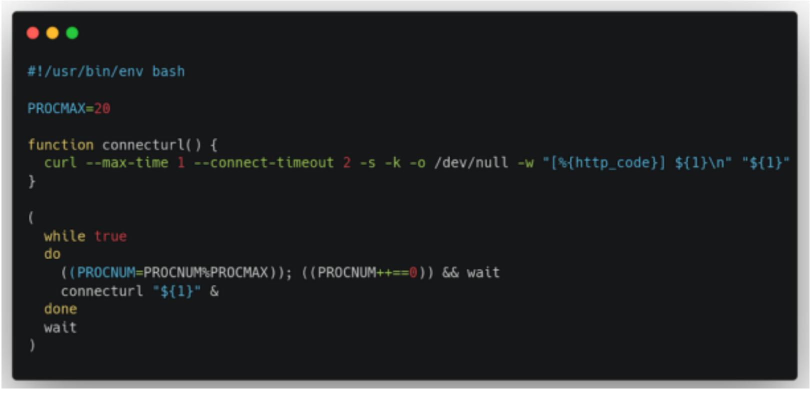
Table 4. The results of brute force attack testing

No	Brute Force Attack Testing (Nmap)	Login Attempts	Hardened Server	Standard Server
1	Test 1	11 Times	Failed	Successful
2	Test 2	15 Times	Failed	Successful
3	Test 3	20 Times	Failed	Successful
4	Test 4	25 Times	Failed	Successful
5	Test 5	30 Times	Failed	Successful

Web Scanning

Web scanning is a cyberattack that uses automated tools or scripts called 'web scanners' to identify and explore vulnerabilities in web applications or websites. Web scanners aim to discover security gaps, such as program weaknesses, insecure server configurations, or software vulnerabilities, which attackers can exploit.

Web scanning attack testing aims to test the resilience and security of endpoints and web services on the target servers, namely the hardened and standard servers. In this scenario, the Attacker attempts to simulate an Active Web Scanning attack using a Custom Tool, as seen in Figure 6. Figure 7(a) shows the results of testing a web scanning attack on a hardened server. The results show that most requests return an HTTP status code of 200, which means the request was successful and the resource is available and accessible. However, some requests returned a response code of 000, indicating that the request was unsuccessful and that the server did not respond. This indicates that the server has configured rate limiting or has other defense mechanisms to protect against web scanning attacks. Figure 7(b) shows the results of testing a web scanning attack on a standard server, which shows that all requests receive a response with an HTTP status code of `200`. This indicates that the server provides a successful response and that the scanned endpoints are accessible without any restrictions or blocking mechanisms detected by the scanning process. Figure 7(b) shows the results of testing a web scanning attack on a standard server, where all requests receive a response with an HTTP status code of 200. This indicates that the server provides a successful response, and all scanned endpoints are accessible without any restrictions or blocking mechanisms detected during the scanning process. The results of the web scanning attack series are presented in Table 5. The data shows that the hardened server successfully limits access to the web server with an average success rate of 34 or 67% and a failure rate of 16 or 33%. On the other hand, the standard server has no access restrictions on the web server, making it vulnerable to web scanning attacks with an average success rate of 100% and a failure rate of 0%. These results indicate that the hardened server effectively repels web scanning attacks compared to the standard server, which does not implement rate limiting or has other defense mechanisms in the web server configuration.



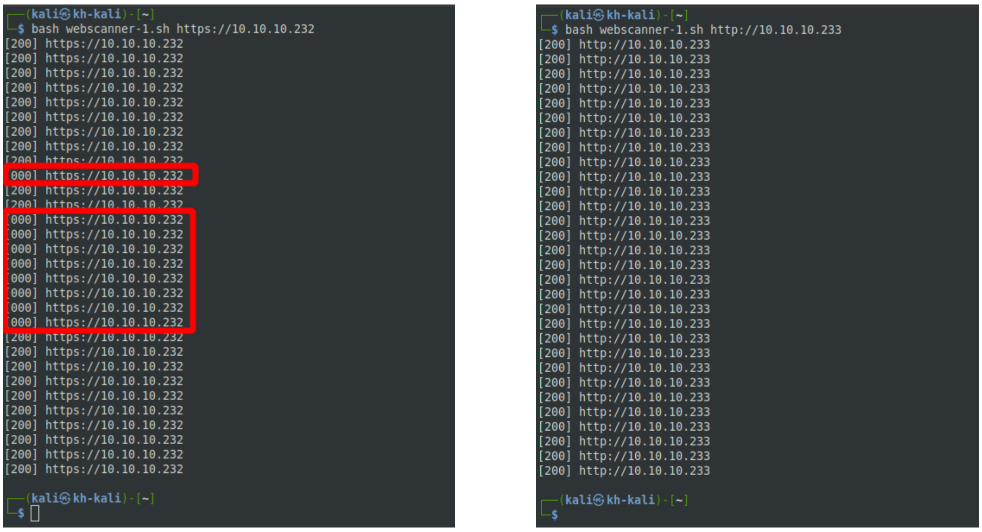
```
#!/usr/bin/env bash

PROCMAX=20

function connecturl() {
  curl --max-time 1 --connect-timeout 2 -s -k -o /dev/null -w "[%{http_code}] ${1}\n" "${1}"
}

(
  while true
  do
    ((PROCNUM=PROCNUM%PROCMAX)); ((PROCNUM++==0)) && wait
    connecturl "${1}" &
  done
  wait
)
```

Figure 5. Web scanning attack script



```
┌──(kali㉿kh-kali)-[~]
└─$ bash webscanner-1.sh https://10.10.10.232
[200] https://10.10.10.232
[200] https://10.10.10.232
[200] https://10.10.10.232
[200] https://10.10.10.232
[200] https://10.10.10.232
[200] https://10.10.10.232
[200] https://10.10.10.232
[200] https://10.10.10.232
[200] https://10.10.10.232
[000] https://10.10.10.232
[200] https://10.10.10.232
[200] https://10.10.10.232
[000] https://10.10.10.232
[000] https://10.10.10.232
[000] https://10.10.10.232
[000] https://10.10.10.232
[000] https://10.10.10.232
[000] https://10.10.10.232
[000] https://10.10.10.232
[000] https://10.10.10.232
[200] https://10.10.10.232
[200] https://10.10.10.232
[200] https://10.10.10.232
[200] https://10.10.10.232
[200] https://10.10.10.232
[200] https://10.10.10.232
[200] https://10.10.10.232
[200] https://10.10.10.232
[200] https://10.10.10.232
[200] https://10.10.10.232

┌──(kali㉿kh-kali)-[~]
└─$
```

(a)

```
┌──(kali㉿kh-kali)-[~]
└─$ bash webscanner-1.sh http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233
[200] http://10.10.10.233

┌──(kali㉿kh-kali)-[~]
└─$
```

(b)

Figure 6. Comparison of Web Scanning attack results on (a) the hardened server and (b) the standard server.

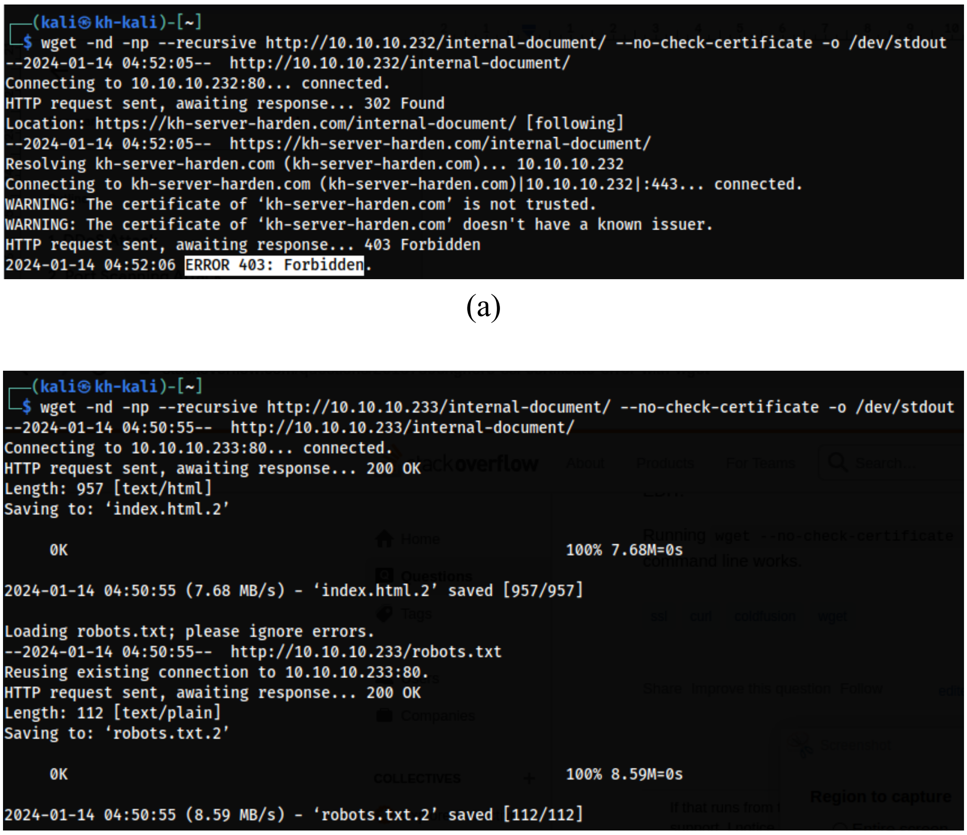
Table 5. The results of web scanning attack testing

No	Web Scanning Attack Testing	Request Attempts	Hardened Server		Standard Server	
			Successful	Failed	Successful	Failed
1	Test 1	30 Times	20	10	30	0
2	Test 2	40 Times	24	16	40	0
3	Test 3	50 Times	34	16	50	0
4	Test 4	60 Times	40	20	60	0
5	Test 5	70 Times	51	19	70	0

Web Crawling

Web crawling attack refers to the crawling activity conducted by bots or automated scripts to gather information from websites. While most web crawling activities are carried out by robots or scripts (such as search engine indexing), there are cases where crawling can be abused for harmful purposes, such as extracting personal data from websites, which can be considered an attack.

Web crawling attack testing is conducted to evaluate and identify vulnerable points and unintentionally exposed resources on the target servers. In this scenario, the Attacker attempts to perform web crawling, which is necessary to duplicate resources on the website so that the Attacker can obtain necessary information. This is done using the WGET tool. Figure 8(a) shows the results of a web crawling attack test on the harden server. Initially, an HTTP request gets a 302 Found response, indicating a transmission, and is then redirected to a new URL using HTTPS. After ignoring the warning and continuing the request, the server returns a 403 Forbidden response, indicating that access to the resource is not by the server. Figure 8(b) shows the results of a web crawling attack test on a standard server, showing that a request using wget successfully accesses and downloads the 'index.html' and 'robots.txt' files from the 'internal-document' directory. Both files were successfully downloaded with an HTTP status of `200 OK`, indicating that no one has enabled access and the web crawler can freely retrieve the files. The results of a series of web crawling attacks are presented in Table 6. The data reveals that the hardened server can block all access attempts with a success rate of 0% for the attack. In contrast, every attack test on the standard server successfully accesses the internal folder on the server with a success rate of 100%. These results show that the hardened server effectively counteracts web crawling attacks compared to the standard server, which does not provide access to internal folders on the web server.



```
┌──(kali㉿kh-kali)-[~]
└─$ wget -nd -np --recursive http://10.10.10.232/internal-document/ --no-check-certificate -o /dev/stdout
--2024-01-14 04:52:05--  http://10.10.10.232/internal-document/
Connecting to 10.10.10.232:80... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://kh-server-harden.com/internal-document/ [following]
--2024-01-14 04:52:05--  https://kh-server-harden.com/internal-document/
Resolving kh-server-harden.com (kh-server-harden.com)... 10.10.10.232
Connecting to kh-server-harden.com (kh-server-harden.com)|10.10.10.232|:443... connected.
WARNING: The certificate of ‘kh-server-harden.com’ is not trusted.
WARNING: The certificate of ‘kh-server-harden.com’ doesn't have a known issuer.
HTTP request sent, awaiting response... 403 Forbidden
2024-01-14 04:52:06 ERROR 403: Forbidden.
```

(a)

```
┌──(kali㉿kh-kali)-[~]
└─$ wget -nd -np --recursive http://10.10.10.233/internal-document/ --no-check-certificate -o /dev/stdout
--2024-01-14 04:50:55--  http://10.10.10.233/internal-document/
Connecting to 10.10.10.233:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 957 [text/html]
Saving to: ‘index.html.2’

     0K                                                  100% 7.68M=0s

2024-01-14 04:50:55 (7.68 MB/s) - ‘index.html.2’ saved [957/957]

Loading robots.txt; please ignore errors.
--2024-01-14 04:50:55--  http://10.10.10.233/robots.txt
Reusing existing connection to 10.10.10.233:80.
HTTP request sent, awaiting response... 200 OK
Length: 112 [text/plain]
Saving to: ‘robots.txt.2’

     0K                                                  100% 8.59M=0s

2024-01-14 04:50:55 (8.59 MB/s) - ‘robots.txt.2’ saved [112/112]
```

(b)

Figure 7. Comparison of Web Scanning attack results on (a) the hardened server and (b) the standard server.

Table 6. The results of web crawling attack testing

No	Web Crawling Attack Testing	URL	Hardened Server	Standard Server
1	Test 1	/internal-document/	403 Forbidden	200 OK
2	Test 2	/confidential/	403 Forbidden	200 OK
3	Test 3	/admin/	403 Forbidden	200 OK
4	Test 4	/user/	403 Forbidden	200 OK
5	Test 5	/backup/	403 Forbidden	200 OK

CONCLUSION

This study demonstrates that applying the CIS Benchmark significantly enhances the security of Ubuntu Server 22.04 by reducing vulnerabilities and increasing resilience against various cyber threats such as DDoS, port scanning, brute force, web scanning, and web crawling, with hardened servers showing notably lower CPU usage during attacks. To extend these benefits, future research should explore broader implementation of CIS guidelines across diverse operating systems and environments, including cloud, containers, and emerging technologies like edge computing, IoT, and AI-driven security tools. Evaluating scalability and effectiveness in complex, real-world, large-scale networks, along with integrating automated compliance and continuous monitoring, will support the development of more adaptive and robust server hardening strategies that address evolving cybersecurity challenges.

REFERENCES

Angione, F., Bernardi, P., Cantoro, R., Giardino, N. D. G., Piumatti, D., Reorda, M. S., Appello, D., & Tancorre, V. (2023). On the integration and hardening of Software Test Libraries in Real-Time Operating Systems. *2023 IEEE 24th Latin American Test Symposium (LATS)*, 1–6.

Bachras, M. (2020). *Supporting the development of infrastructure as code using Ansible: a smart IDE integrating external sources*.

Comer, D. (2025). *Operating system design: The Xinu approach*. CRC Press.

Deriyanto, S. P., & Santoso, H. A. (2020). Development of Bot for Microservices Server Monitoring Using Life Cycle Approach to Network Design Method. *JUITA: Jurnal Informatika*, *8*(2), 141–147.

Dieber, B., Breiling, B., Taurer, S., Kacianka, S., Rass, S., & Schartner, P. (2017). Security for the robot operating system. *Robotics and Autonomous Systems*, *98*, 192–203.

Ernawati, R., Ruslianto, I., & Bahri, S. (2022). Implementasi metode port knocking pada sistem keamanan server ubuntu virtual berbasis web monitoring. *Coding Jurnal Komputer Dan Aplikasi*, *10*(01), 158–169.

Fitriani, R., Subagiyo, R., & Asiyah, B. N. (2023). Mitigating IT Risk of Bank Syariah Indonesia: A Study of Cyber Attack on May 8, 2023. *Al-Amwal: Jurnal Ekonomi Dan Perbankan Syari'ah*, *15*(1), 86–100.

Hamidy, F., & Yasin, I. (2024). Penerapan Metode Moving Average Dalam Penentuan Harga Pokok Penjualan Barang Berbasis Web. *CHAIN: Journal of Computer Technology, Computer Engineering, and Informatics*, *2*(2), 67–76.

Hyppönen, M. (2021). Securing a linux server against cyber attacks. *Mestrado, Tampere University, Tampere*.

Ilonen, T., Virtanen, S., & Isoaho, J. (2024). *Operating System Hardening Based on Privacy, Security and Performance: Customization of Microsoft Windows*.

Irawan, H., Paamsyah, J., Feprizon, H., & Fatullah, A. P. (2024). Pengaturan Tindak Pidana Mayantara (Cybercrime) Dalam Sistem Hukum Indonesia. *Innovative: Journal Of Social Science Research*, *4*(1), 4358–4369.

Irfandi, F. R., Kurnia, Y., Hedianto, S., & Almaarif, A. (2022). Software Security Hardening Pada Virtual Private Server Berdasarkan NIST SP 800-123 di Universitas XYZ. *J. Inf. Syst. Res*, *4*(1), 94–102.

Jaeger, T. (2022). *Operating system security*. Springer Nature.

Leiritie, T. (2023). *Automated hardening of Linux infrastructure*.

Nieminen, R. (2025). *Windowsin koventaminen CIS-CAT Lite palvelulla*.

Prastika, D. P., Triyono, J., & Lestari, U. (2019). *Audit Dan Implementasi Cis Benchmark Pada Sistem Operasi Linux Debian Server (Studi Kasus: Server Laboratorium Jaringan Dan Komputer 6 Institut Sains & Teknologi Akprind Yogyakarta)*.

Sari, I. P., Pasaribu, I. W., As, M. Z. A., & Octanelsha, B. C. (2024). Analisis Kebijakan Cyber Crime Dalam Hukum Positif Di Indonesia. *Journal Of Law And Nation*, *3*(2), 395–403.

Stallings, W. (2018). *Operating Systems: Internals and Design Principles, 9/e*. Pearson IT Certification.

Suhaemin, A., & Muslih, M. (2021). Karakteristik cybercrime di indonesia. *Edulaw: Journal of Islamic Law and Jurisprudance*, *2*(1), 15–26.

Sutanto, A., Khakim, L., Hartono, E. B., & Umam, M. K. (2025). Enhancing Vocational School Students'competence In Operating System Security Through Hardening Training To Improve Awareness Of Cyber Threats. *Jmm (Jurnal Masyarakat Mandiri)*, *9*(1), 978–989.

Tevault, D. A. (2020). *Mastering Linux Security and Hardening: Protect your Linux systems from intruders, malware attacks, and other cyber threats*. Packt Publishing Ltd.

Tevault, D. A. (2023). *Mastering Linux Security and Hardening: A practical guide to protecting your Linux system from cyber attacks*. Packt Publishing Ltd.

Vakhula, O., Kurii, Y., Opirskyy, I., & Susukailo, V. (2024). Security as Code Concept for Fulfilling ISO/IEC 27001: 2022 Requirements. *CPITS*, 59–72.

Wijaya, C. C., & Budiman, A. S. (2023). Perancangan Keamanan Jaringan Komputer Pada Router Dengan Metode ACL Pada PT. Aruna Sinar Jaya Jakarta. *Journal Zetroem*, *5*(2), 180–186.