

Cyber report 1

Executive Summary:

- Mass of IPs were identified in the `/var/log/apache2/` so we had to take some serious actions about it
- Cloud flare WAF was added
- More on security hardening in the "[Security/ Security Incident & Environment Hardening Report](#)"
- More than one IP address got identified from countries like

· Country / Region	· Traffic
· Sweden	· 87,973
· United States	· 110
· United Kingdom	· 99
· Germany	· 98
· Egypt	· 25

- All the traffic was a big red flag since the site isn't published yet (except EG)
- After further investigation we know that they were a web crawlers / Ai bot and some of them were pen testing

Mitigation:

1. **Cloudflare WAF**
2. **.htaccess**

Process From the start till mitigation:

IPs requests on the apache server :

After getting the domain thoutha.page a flood of requests appeared in the logs

```
537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36"
www.thoutha.page:80 35.238.96.173 - - [30/Nov/2025:12:27:29 +0000] "GET / HTTP/1.1" 301 574 "-" "python-requests/2.32.5"
www.thoutha.page:80 92.113.7.16 - - [30/Nov/2025:12:30:34 +0000] "GET /.env HTTP/1.1" 301 545 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.140 Safari/537.36"
www.thoutha.page:80 92.113.7.16 - - [30/Nov/2025:12:30:34 +0000] "POST / HTTP/1.1" 301 537 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.140 Safari/537.36"
www.thoutha.page:80 165.232.151.244 - - [30/Nov/2025:12:41:45 +0000] "GET /wp-login.php HTTP/1.1" 301 597 "-" "Mozilla/5.0"
www.thoutha.page:80 29.64.104.143 - - [30/Nov/2025:12:45:01 +0000] "GET /developmentserver/metadataparser HTTP/1.1" 301 586 "-" "Mozilla/5.0 zgrab/0.x"
www.thoutha.page:80 54.226.107.87 - - [30/Nov/2025:12:51:24 +0000] "GET / HTTP/1.1" 301 518 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36"
www.thoutha.page:80 285.210.31.27 - - [30/Nov/2025:12:52:15 +0000] "GET /.well-known/security.txt HTTP/1.1" 301 566 "-" "Hello from Palo Alto Networks, find out more about our scans in https://docs.cortex.paloaltonetworks.com/r/1/cortex-xpanse/scanning-activity"
www.thoutha.page:80 112.46.215.180 - - [30/Nov/2025:12:57:30 +0000] "GET / HTTP/1.1" 400 490 "-" "-"
www.thoutha.page:80 43.153.122.30 - - [30/Nov/2025:12:58:10 +0000] "GET / HTTP/1.1" 301 537 "-" "Mozilla/5.0 (iPhone; CPU iPhone OS 13_2_3 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/13.0.3 Mobile/15E148 Safari/604.1"
www.thoutha.page:80 45.55.130.218 - - [30/Nov/2025:13:28:20 +0000] "GET / HTTP/1.1" 301 574 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36"
www.thoutha.page:80 45.55.130.218 - - [30/Nov/2025:13:28:20 +0000] "GET /favicon.ico HTTP/1.1" 301 596 "http://13.49.221.187/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36"
www.thoutha.page:80 181.32.218.31 - - [30/Nov/2025:14:05:50 +0000] "HEAD /Core/Skin/Login.aspx HTTP/1.1" 301 205 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.5845.140 Safari/537.36"
www.thoutha.page:80 184.164.158.246 - - [30/Nov/2025:14:29:25 +0000] "GET / HTTP/1.1" 301 573 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36"
www.thoutha.page:80 79.124.40.84 - - [30/Nov/2025:15:00:53 +0000] "HEAD / HTTP/1.0" 301 204 "-" "-"
www.thoutha.page:80 187.172.116.148 - - [30/Nov/2025:15:02:18 +0000] "GET /.env HTTP/1.1" 301 545 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.140 Safari/537.36"
www.thoutha.page:80 187.172.116.148 - - [30/Nov/2025:15:02:18 +0000] "POST / HTTP/1.1" 301 537 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.140 Safari/537.36"
www.thoutha.page:80 35.244.6.162 - - [30/Nov/2025:15:19:22 +0000] "GET / HTTP/1.1" 301 574 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3"
www.thoutha.page:80 93.174.93.12 - - [30/Nov/2025:15:21:12 +0000] "\x16\x03\x02\x01\x01" 400 490 "-" "-"
www.thoutha.page:80 209.38.201.154 - - [30/Nov/2025:15:33:08 +0000] "" 400 490 "-" "-"
www.thoutha.page:80 35.244.6.162 - - [30/Nov/2025:15:45:00 +0000] "GET / HTTP/1.1" 301 574 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3"
www.thoutha.page:80 181.32.218.31 - - [30/Nov/2025:15:45:48 +0000] "HEAD /Core/Skin/Login.aspx HTTP/1.1" 301 205 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36"
www.thoutha.page:80 209.38.201.154 - - [30/Nov/2025:15:47:35 +0000] "GET /robots.txt HTTP/1.1" 301 557 "-" "Mozilla/5.0"
www.thoutha.page:80 34.85.243.41 - - [30/Nov/2025:15:52:29 +0000] "GET / HTTP/1.1" 301 517 "-" "Mozilla/5.0 (compatible; CMS-Checker/1.0; +https://example.com)"
www.thoutha.page:80 192.238.247.144 - - [30/Nov/2025:15:52:47 +0000] "GET / HTTP/1.1" 301 574 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.100 Safari/537.36"
www.thoutha.page:80 35.244.6.162 - - [30/Nov/2025:15:58:01 +0000] "GET / HTTP/1.1" 301 574 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.3029.110 Safari/537.3"
```

Figure 1

Findings

As shown in Figure 1, most inbound requests were attempting to GET sensitive directories and configuration files, including /.env, /.git/config, .vscode/sftp.json, .DS_Store, /config.json, /server-status, /v2/_catalog, /login.action, and /wp-login.php.

These paths are commonly targeted during the reconnaissance and enumeration phases of penetration testing and malicious scanning operations.

Requests originated from a wide range of distributed IPs and leveraged automated scanners such as L9scan, zgrab, and Cortex Xpanse, indicating systematic probing rather than normal user traffic.

From a penetration testing perspective, the activity aligns with typical directory brute-forcing, endpoint enumeration, and configuration leakage attempts, where attackers aim to extract secrets, identify technology stacks, or locate misconfigurations such as exposed Git repositories, environment variables, or debugging interfaces.

Repeated attempts to access WordPress login pages although we don't have WordPress, Exchange-related export endpoints, and internal metadata paths suggest opportunistic scanning for known high-impact vulnerabilities and misconfigured services.

Although no successful exploitation was observed based on HTTP response patterns, the frequency and diversity of these probes confirm that the server is continuously targeted by broad scanning campaigns.

This reinforces the importance of maintaining a hardened perimeter and enforcing proactive detection controls.

Mitigation

1. Web Server Hardening

- Block access to sensitive files (e.g., .env, .git, .DS_Store, .vscode) using .htaccess.
- Disable directory listing on all virtual hosts.
- With adding **Blocking** rules from CloudFlare for extra layer of protection as shown in Figure 2:

Custom rules 3/5 used Create rule ⚡ Summarize with Cloudflare						Go to detection settings
Order	Name	Match against	Action	CSR ⓘ	Events last 24h	
1	AI Crawl Control - Block AI bots by User Agent	URI Path does not equal /robots.txt, User Agent contains archive.org_bot, User Agent contains bingbot, User Age...	Block	-	 1	Active ⋮
2	Block	URI Path contains .env or URI Path contains /env or URI Path contains /etc/passwd or URI Path contains...	Block	-	 1	Active ⋮
3	Geo Block	Country does not equal EG	Block	-	 0	Active ⋮

Figure 2

2. Access Control and Network Filtering

- Enforce Web Application Firewall (WAF) rules to filter requests to sensitive or invalid paths.
- Use Geoloc filtering if the service does not require global accessibility as shown in the tests in Figure 3&4.
- Apply rate limiting to reduce automated probing.

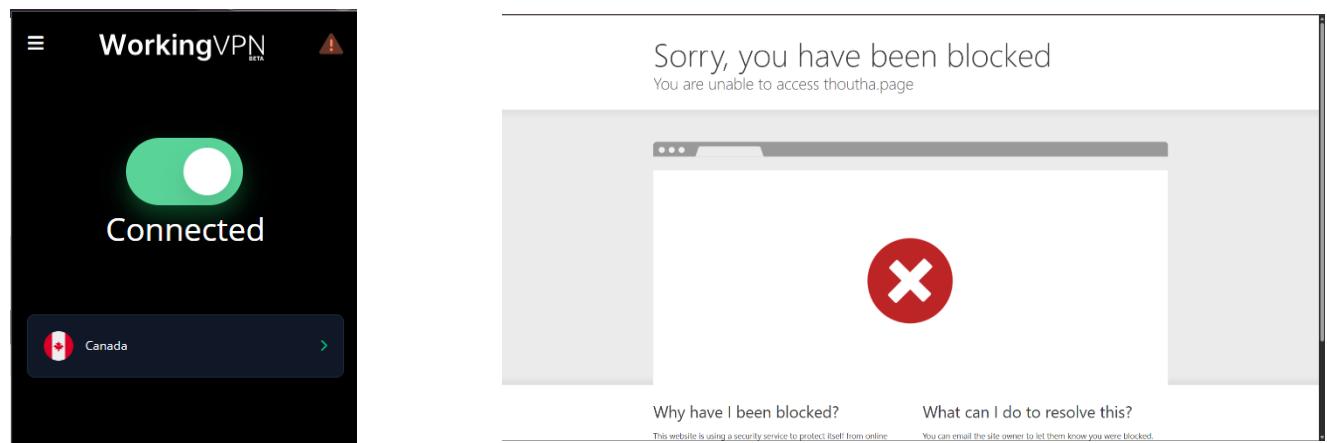


Figure 4

Figure 3

3. Surface Reduction and Configuration Security

- Remove or relocate development files such as .vscode, .DS_Store, debug panels, and test endpoints storing them back in the dev folder in the home dir.
- Ensure that no environment variables or secret keys are exposed externally.
- Confirm that repository directories (e.g., .git) are never accessible through the web root.

4. Logging, Monitoring, and Detection

- Enable detailed access and error logging with log rotation.

5. Vulnerability Management

- Maintain continuous external attack surface monitoring and performing demo penetration test on what's done on the server to check for opened ports and services for now although they aren't much of them like in the Figure 5.

```
~ > joseph@outis ~ -> nmap 13.49.221.187
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-30 22:51 EST
Nmap scan report for ec2-13-49-221-187.eu-north-1.compute.amazonaws.com (13.49.221.187)
Host is up (0.023s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https
3389/tcp  open  ms-wbt-server
5000/tcp  open  upnp
8080/tcp  open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 49.08 seconds

~ > joseph@outis ~ -> nmap thouha.page
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-30 22:53 EST
Stats: 0:00:01 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 3.00% done; ETC: 22:53 (0:00:32 remaining)
Nmap scan report for thouha.page (104.21.61.135)
Host is up (0.016s latency).
Other addresses for thouha.page (not scanned): 172.67.210.165 2606:4700:3035::6815:3d87 2606:4700:3031::ac43:d2a5
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
8080/tcp  open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 5.83 seconds
```

Figure 5

- Keep all server components, frameworks, and plugins fully patched.

Future plans and further mitigation:

- Integrate logs with a SIEM for correlation and alerting.
- Detect repeated scanning behavior or known scanner signatures.