

In HIS name

Ali Sheikh Attar

Nmap & ebpf security tools demo

- Nmap

For this demo i use my home router network

Network Inventory & Discovery

to identify all devices connected to a network

```
nmap -sn 192.168.1.0/24
```

This command performs a "ping scan" (-sn), which tells Nmap to only discover hosts without port scanning. It scans the entire subnet 192.168.1.0/24 to list all active devices.

```
asa@ASAttar-ASUS:~$ nmap -sn 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-07-28 13:08 +0330
Nmap scan report for _gateway (192.168.1.1)
Host is up (0.0085s latency).
Nmap scan report for ASAttar-ASUS (192.168.1.4)
Host is up (0.00015s latency).
Nmap scan report for 192.168.1.5
Host is up (0.060s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 3.08 seconds
```

output

Vulnerability Assessment

check if any devices on the network are vulnerable to a known exploit.

```
nmap --script vuln 192.168.1.0/24
```

Explanation: The --script vuln option tells Nmap to run a series of vulnerability detection scripts against the target network 192.168.1.0/24. This helps identify common vulnerabilities.

```

asa@ASAttar-ASUS:~$ nmap --script vuln 192.168.1.8/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-07-28 14:35 +0330
Nmap scan report for _gateway (192.168.1.1)
Host is up (0.025s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
|_sslv2-drown:
22/tcp    open  ssh
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
23/tcp    open  telnet
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
80/tcp    open  http
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
| http-cookie-flags:
|   /:
|     SessionID:
|       httponly flag not set
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-litespeed-sourcecode-download:
| Litespeed Web Server Source Code Disclosure (CVE-2010-2333)
|/index.php source code:
|<html><head><meta HTTP-EQUIV="Pragma" CONTENT="no-cache"><script language='javascript'>parent.location="/login.htm"</script></head><body></body></html>
|_http-majordomo2-dir-traversal: ERROR: Script execution failed (use -d to debug)
| http-phpmyadmin-dir-traversal:
|   VULNERABLE:
|     phpMyAdmin grab_globals.lib.php subform Parameter Traversal Local File Inclusion
|       State: UNKNOWN (unable to test)
|       IDs: CVE:2005-3299
|         PHP file inclusion vulnerability in grab_globals.lib.php in phpMyAdmin 2.6.4 and 2.6.4-pl1 allows remote attackers to include local files via the $_
|           Disclosure date: 2005-10-11
|           Extra information:
|             ../../../../../../etc/passwd :
|             <html><head><meta HTTP-EQUIV="Pragma" CONTENT="no-cache"><script language='javascript'>parent.location="/login.htm"</script></head><body></body></html>
|               References:
|                 http://www.exploit-db.com/exploits/1244/
|                 https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3299
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-vuln-cve2017-1001000: ERROR: Script execution failed (use -d to debug)
5431/tcp open  park-agent
|_clamav-exec: ERROR: Script execution failed (use -d to debug)

Nmap scan report for ASAttar-ASUS (192.168.1.4)
Host is up (0.000047s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
80/tcp    open  http
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-enum:
|   /server-status/: Potentially interesting folder
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.

Nmap done: 256 IP addresses (2 hosts up) scanned in 36.59 seconds

```

Output

Review of output

- Common Errors:

Many instances of the clamav-exec script failing. This could be due to the script not being properly installed or configured, or missing dependencies.

- Specific Findings:

- 192.168.1.1 (Gateway):
 - http-cookie-flags: Missing httponly flag on SessionID, which is a minor security issue as it might expose the session ID to client-side scripts.

- Litespeed Web Server Source Code Disclosure (CVE-2010-2333): This indicates a potential serious vulnerability where the server's source code can be disclosed.
- phpMyAdmin Local File Inclusion (CVE-2005-3299): Potential vulnerability that could allow an attacker to include local files.
- 192.168.1.4 (ASAttar-ASUS):
 - /server-status/: Directory found which may contain server status information that could be useful for further attacks.
- Recommendations:
 - Fix Vulnerabilities:
 - Apply patches and updates for the identified vulnerabilities, especially the Litespeed Web Server Source Code Disclosure and phpMyAdmin Local File Inclusion.
 - Configuration Improvements:
 - Set the `httponly` flag on cookies to enhance security.
- Script Execution Issues:
 - Investigate why the `clamav-exec` script is failing. This might involve checking for proper installation and configuration of ClamAV and its dependencies.

Port Scanning

determine which ports are open on devices within the network.

```
nmap --script vuln 192.168.1.0/24
```

Explanation: This command scans all 65,535 ports on all devices within the network range 192.168.1.0/24. It helps in identifying services running on non-standard ports.

```
asa@ASAttar-ASUS:~$ nmap --script vuln 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-07-28 14:50 +0330
Nmap scan report for _gateway (192.168.1.1)
Host is up (0.026s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
|_sslv2-down:
22/tcp    open  ssh
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
23/tcp    open  telnet
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
80/tcp    open  http
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
| http-cookie-flags:
|_ /:
```

Output for 192.168.1.1 -> 4 open ports

```
Nmap scan report for ASAttar-ASUS (192.168.1.4)
Host is up (0.000061s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
80/tcp    open  http
|_clamav-exec: ERROR: Script execution failed (use -d to debug)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-enum:
|_ /server-status/: Potentially interesting folder
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.

Nmap done: 256 IP addresses (2 hosts up) scanned in 36.52 seconds
```

Output for 192.168.1.4 -> 1 open ports

OS and Service Detection

identifying the operating systems and services running on devices within the network.

```
nmap -A 192.168.1.0/24
```

Explanation: The -A flag enables OS detection, version detection, script scanning, and traceroute, providing comprehensive information about devices within the network range 192.168.1.0/24.

```
Nmap scan report for _gateway (192.168.1.1)
Host is up (0.024s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp
| fingerprint-strings:
|   GenericLines, SMBProgNeg, SSLSessionReq:
|     220 FTP server (192.168.1.1) ready.
|     Unknown command: ""
| Help:
|     220 FTP server (192.168.1.1) ready.
|     USER expected.
| NULL:
|     220 FTP server (192.168.1.1) ready.
22/tcp    open  ssh      Dropbear sshd 0.48 (protocol 2.0)
| ssh-hostkey:
|_ 1040 d0:0d:e9:4e:24:4c:41:86:e9:62:32:5c:24:de:3e:4f (RSA)
23/tcp    open  telnet
```

Output for gateway

```
Nmap scan report for ASAttar-ASUS (192.168.1.4)
Host is up (0.000061s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.52 ((Ubuntu))
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
```

Output for my laptop

Firewall and IDS Evasion

perform a stealth scan to avoid detection by firewalls or Intrusion Detection Systems (IDS).

```
nmap -sS -T0 -Pn 192.168.1.0/24
```

Explanation: The -sS option performs a stealth SYN scan, -T0 sets the timing to "paranoid" to slow down the scan, and -Pn tells Nmap to skip the ping check. This makes the scan less likely to be detected.

Note

this type of scan requires root privilege

```
asa@ASAttar-ASUS:~$ nmap -sS -T0 -Pn 192.168.1.0/24
You requested a scan type which requires root privileges.
QUITTING!
```

```
asa@ASAttar-ASUS:~$ sudo nmap -sS -T2 -Pn 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-07-28 15:22 +0330
Stats: 0:00:02 elapsed; 0 hosts completed (0 up), 255 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 2.35% done; ETC: 15:24 (0:02:05 remaining)
Stats: 0:00:04 elapsed; 0 hosts completed (0 up), 255 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 3.92% done; ETC: 15:24 (0:01:38 remaining)
Stats: 0:00:04 elapsed; 0 hosts completed (0 up), 255 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 4.12% done; ETC: 15:24 (0:01:56 remaining)
Stats: 0:00:06 elapsed; 0 hosts completed (0 up), 255 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 5.49% done; ETC: 15:24 (0:01:43 remaining)
Nmap scan report for _gateway (192.168.1.1)
Host is up (0.0040s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http
5431/tcp  open  park-agent
MAC Address: BC:34:00:53:BC:E6 (Ieee Registration Authority)

Nmap scan report for ASAttar-ASUS (192.168.1.4)
Host is up (0.000086s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 256 IP addresses (2 hosts up) scanned in 990.07 seconds
```

Output

Note

I use T2 because of time issues

Scripted Interaction with Hosts

Scenario: checking the SSL certificate of web servers on the network for expiry.

```
nmap --script ssl-cert -p 443 192.168.1.0/24
```

Explanation: The --script ssl-cert option runs the ssl-cert NSE script on port 443 of devices within the network range 192.168.1.0/24, retrieving information about their SSL certificates, including expiry dates.

```
asa@ASAttar-ASUS:~$ nmap --script ssl-cert -p 443 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-07-28 16:08 +0330
Nmap scan report for _gateway (192.168.1.1)
Host is up (0.015s latency).

PORT      STATE SERVICE
443/tcp    closed https

Nmap scan report for 192.168.1.2
Host is up (0.029s latency).

PORT      STATE SERVICE
443/tcp    closed https

Nmap scan report for ASAttar-ASUS (192.168.1.4)
Host is up (0.000078s latency).

PORT      STATE SERVICE
443/tcp    closed https

Nmap done: 256 IP addresses (3 hosts up) scanned in 3.21 seconds
```

output - the 443 ports are closed for three hosts in network including gateway

Checking for Common Malware Infections

investigating devices for signs of common malware infections.

```
nmap --script http-malware-host 192.168.1.0/24
```

Explanation: The --script http-malware-host option uses a specific Nmap script to check if devices within the network range 192.168.1.0/24 are associated with known malware.

```
asa@ASAttar-ASUS:~$ nmap --script http-malware-host 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-07-28 16:15 +0330
Nmap scan report for _gateway (192.168.1.1)
Host is up (0.029s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http
|_http-malware-host: false
5431/tcp  open  park-agent

Nmap scan report for 192.168.1.2
Host is up (0.0087s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
49152/tcp open  unknown
62078/tcp open  iphone-sync

Nmap scan report for ASAttar-ASUS (192.168.1.4)
Host is up (0.000057s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
80/tcp    open  http
|_http-malware-host: Host appears to be clean

Nmap done: 256 IP addresses (3 hosts up) scanned in 15.27 seconds
```

Output

3 hosts are up in the network which the http-malware-host flag is flase or appears to be clean , which makes our network clean.

Network Performance and Connectivity Testing

testing the performance and connectivity of a network path.

```
nmap --traceroute 192.168.1.0/24
```

Explanation: The --traceroute option provides information about the path packets take to reach devices within the network range 192.168.1.0/24, which is useful for diagnosing network performance issues.

In summary, Nmap's traceroute function effectively maps the path and counts the intermediary devices (hops) from your device to the target IPs.

```
asa@ASAttar-ASUS:~$ nmap --traceroute 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-07-28 18:41 +0330
Traceroute has to be run as root
QUITTING!
```

Note

The –traceroute option need router privilege to be used

```
asa@ASAttar-ASUS:~$ sudo nmap --traceroute 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-07-28 18:43 +0330
Nmap scan report for _gateway (192.168.1.1)
Host is up (0.061s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http
5431/tcp  open  park-agent
MAC Address: BC:34:00:53:BC:E6 (IEEE Registration Authority)

TRACEROUTE
HOP RTT      ADDRESS
1  60.89 ms  _gateway (192.168.1.1)

Nmap scan report for 192.168.1.5
Host is up (0.0065s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
49152/tcp open  unknown
62078/tcp open  iphone-sync
MAC Address: 72:41:7B:BC:60:B2 (Unknown)

TRACEROUTE
HOP RTT      ADDRESS
1  6.48 ms  192.168.1.5

Nmap scan report for ASAttar-ASUS (192.168.1.4)
Host is up (0.0000030s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 256 IP addresses (3 hosts up) scanned in 8.15 seconds
```

output

- The gateway 192.168.1.1 is confirmed as the router since all devices have it as their first hop.
- The latency (RTT) for reaching the gateway and other devices is relatively low, indicating a local network scan.

Identifying Backdoor Services

checking devices for backdoor services that might have been installed by an attacker.

```
nmap -sV --script=backdoor 192.168.1.0/24
```

Explanation: The -sV option performs version detection, and --script=backdoor runs scripts designed to detect backdoor services on devices within the network range 192.168.1.0/24.

```

asa@ASAttar-ASUS:~$ nmap -sV --script '*backdoor*' 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-07-28 19:12 +0330
Stats: 0:00:02 elapsed; 255 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 97.00% done; ETC: 19:12 (0:00:00 remaining)
Stats: 0:00:09 elapsed; 255 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 0.00% done
Nmap scan report for ASAttar-ASUS (192.168.1.4)
Host is up (0.000068s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.52 ((Ubuntu))
|_http-server-header: Apache/2.4.52 (Ubuntu)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (1 host up) scanned in 9.07 seconds

```

Use wildcards to match all scripts about backdoor.

Nmap did not find any backdoors on the scanned host. This is a positive indication.

Compliance Auditing

verifying that a network is compliant with specific security standards (e.g., PCI-DSS).

```
nmap --script pcilist 192.168.1.0/24
```

Explanation: The --script pcilist runs an Nmap script to check for compliance with PCI-DSS standards across the network range 192.168.1.0/24.

```

asa@ASAttar-ASUS:~$ sudo nmap -sV --script=*vuln*,*auth*,*safe* 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2024-07-28 19:25 +0330
Stats: 0:00:02 elapsed; 0 hosts completed (0 up), 255 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 61.57% done; ETC: 19:25 (0:00:01 remaining)
Stats: 0:00:02 elapsed; 0 hosts completed (0 up), 255 undergoing ARP Ping Scan
ARP Ping Scan Timing: About 76.27% done; ETC: 19:25 (0:00:01 remaining)
Stats: 0:00:22 elapsed; 253 hosts completed (2 up), 2 undergoing Service Scan
Service scan Timing: About 60.00% done; ETC: 19:26 (0:00:11 remaining)
Stats: 0:00:32 elapsed; 253 hosts completed (2 up), 2 undergoing Service Scan
Service scan Timing: About 60.00% done; ETC: 19:26 (0:00:17 remaining)
Stats: 0:00:40 elapsed; 253 hosts completed (2 up), 2 undergoing Service Scan
Service scan Timing: About 80.00% done; ETC: 19:26 (0:00:09 remaining)
Nmap scan report for _gateway (192.168.1.1)
Host is up (0.013s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp
| fingerprint-strings:
|   GenericLines, SMBProgNeg, SSLSessionReq:
|     220 FTP server (192.168.1.1) ready.
|     Unknown command: ""
|   Help:
|     220 FTP server (192.168.1.1) ready.
|     USER expected.
|   NULL:
|     220 FTP server (192.168.1.1) ready.
22/tcp    open  ssh    Dropbear sshd 0.48 (protocol 2.0)
|_ssh-auth-methods: ERROR: Script execution failed (use -d to debug)

```

Run combination of scripts instead of pcilist because i dont have it in scripts

```

CVE-2007-1099 7.5 https://vulners.com/cve/CVE-2007-1099
EDB-ID:40119 6.4 https://vulners.com/exploitdb/EDB-ID:40119 *EXPLOIT*
CVE-2016-3116 6.4 https://vulners.com/cve/CVE-2016-3116
CVE-2023-48795 5.9 https://vulners.com/cve/CVE-2023-48795
PACKETSTORM:136251 5.5 https://vulners.com/packetstorm/PACKETSTORM:136251 *EXPLOIT*
EXPLOITPACK:F92411A645D8SF05BDBD274FD22226F 5.5 https://vulners.com/exploitpack/EXPLOITPACK:F92411A645D8SF05BDBD274FD22226F *EXPLOIT*
CVE-2016-7409 5.5 https://vulners.com/cve/CVE-2016-7409
1337DAY-ID-25388 5.5 https://vulners.com/zdt/1337DAY-ID-25388 *EXPLOIT*
MSF:AUXILIARY-SCANNER-SSH-SH_SSH_ENUMUSERS- 5.3 https://vulners.com/metasploit/MSF:AUXILIARY-SCANNER-SSH-SH_SSH_ENUMUSERS-*EXPLOIT*
EDB-ID:45939 5.3 https://vulners.com/exploitdb/EDB-ID:45939 *EXPLOIT*
EDB-ID:45233 5.3 https://vulners.com/exploitdb/EDB-ID:45233 *EXPLOIT*
CVE-2018-15599 5.3 https://vulners.com/cve/CVE-2018-15599
SSH_ENUM 5.0 https://vulners.com/canvas/SSH_ENUM *EXPLOIT*
PACKETSTORM:150621 5.0 https://vulners.com/packetstorm/PACKETSTORM:150621 *EXPLOIT*
EXPLOITPACK:F957D7E8A0CC1E23C3C649B764E13FB0 5.0 https://vulners.com/exploitpack/EXPLOITPACK:F957D7E8A0CC1E23C3C649B764E13FB0 *EXPLOIT*
EXPLOITPACK:EBDBC5685E3276D648B4D14B75563283 5.0 https://vulners.com/exploitpack/EXPLOITPACK:EBDBC5685E3276D648B4D14B75563283 *EXPLOIT*
CVE-2013-4434 5.0 https://vulners.com/cve/CVE-2013-4434
CVE-2013-4421 5.0 https://vulners.com/cve/CVE-2013-4421
1337DAY-ID-31730 5.0 https://vulners.com/zdt/1337DAY-ID-31730 *EXPLOIT*
CVE-2017-9079 4.7 https://vulners.com/cve/CVE-2017-9079
1337DAY-ID-30937 0.0 https://vulners.com/zdt/1337DAY-ID-30937 *EXPLOIT*
```
nmap -t1nct

```

```

tcp open http Apache httpd 2.4.52 ((Ubuntu))
http-server-header: Apache/2.4.52 (Ubuntu)
vulnerabilities:
 cpe:/a:apache:http_server:2.4.52:
 F607361B-6369-5DF5-9B29-E90FA29DC565 9.8 https://vulners.com/githubexploit/F607361B-6369-5DF5-9B29-E90FA29DC565 *EXPLOIT*
 CVE-2023-25690 9.8 https://vulners.com/cve/CVE-2023-25690
 CVE-2022-31813 9.8 https://vulners.com/cve/CVE-2022-31813
 CVE-2022-23943 9.8 https://vulners.com/cve/CVE-2022-23943
 CVE-2022-22720 9.8 https://vulners.com/cve/CVE-2022-22720
 SC1BB960-90C1-5EBF-9BEF-F58BFFD9E09 9.8 https://vulners.com/githubexploit/SC1BB960-90C1-5EBF-9BEF-F58BFFD9E09 *EXPLOIT*
 3F17CA20-788F-5C45-88B3-E12DB2979B7B 9.8 https://vulners.com/githubexploit/3F17CA20-788F-5C45-88B3-E12DB2979B7B *EXPLOIT*
 1337DAY-ID-39214 9.8 https://vulners.com/zdt/1337DAY-ID-39214 *EXPLOIT*
 CVE-2022-28615 9.1 https://vulners.com/cve/CVE-2022-28615
 CVE-2022-22721 9.1 https://vulners.com/cve/CVE-2022-22721
 CVE-2022-36760 9.0 https://vulners.com/cve/CVE-2022-36760
 PACKETSTORM:176334 7.5 https://vulners.com/packetstorm/PACKETSTORM:176334 *EXPLOIT*
 F7F6E599-CEF4-5E03-8E10-FE18C4101E38 7.5 https://vulners.com/githubexploit/F7F6E599-CEF4-5E03-8E10-FE18C4101E38 *EXPLOIT*
 ESC174E5-D6E8-56E0-8403-D287DE52EB3F 7.5 https://vulners.com/githubexploit/ESC174E5-D6E8-56E0-8403-D287DE52EB3F *EXPLOIT*
 DB6E1BBD-08B1-574D-A351-7D6BB9898A4A 7.5 https://vulners.com/githubexploit/DB6E1BBD-08B1-574D-A351-7D6BB9898A4A *EXPLOIT*
 CVE-2024-27316 7.5 https://vulners.com/cve/CVE-2024-27316
 CVE-2023-31122 7.5 https://vulners.com/cve/CVE-2023-31122
 CVE-2023-27522 7.5 https://vulners.com/cve/CVE-2023-27522
 CVE-2022-30556 7.5 https://vulners.com/cve/CVE-2022-30556
 CVE-2022-29404 7.5 https://vulners.com/cve/CVE-2022-29404
 CVE-2022-26377 7.5 https://vulners.com/cve/CVE-2022-26377
 CVE-2022-22719 7.5 https://vulners.com/cve/CVE-2022-22719
 CVE-2006-20001 7.5 https://vulners.com/cve/CVE-2006-20001
 B0A9E5E8-7CCC-5984-9922-A89F11D6BF38 7.5 https://vulners.com/githubexploit/B0A9E5E8-7CCC-5984-9922-A89F11D6BF38 *EXPLOIT*
 B0208442-6E17-5772-B12D-B5BE30FA5540 7.5 https://vulners.com/githubexploit/B0208442-6E17-5772-B12D-B5BE30FA5540 *EXPLOIT*
 A820A056-9F91-5059-B0BC-8D92C7A31A52 7.5 https://vulners.com/githubexploit/A820A056-9F91-5059-B0BC-8D92C7A31A52 *EXPLOIT*
 A0F268C8-7319-5637-82F7-8DAF72D14629 7.5 https://vulners.com/githubexploit/A0F268C8-7319-5637-82F7-8DAF72D14629 *EXPLOIT*
 9814661A-35A4-5DB7-BB25-A1040F365C81 7.5 https://vulners.com/githubexploit/9814661A-35A4-5DB7-BB25-A1040F365C81 *EXPLOIT*
 5A864BCC-B490-5532-83AB-2E4109BB83C31 7.5 https://vulners.com/githubexploit/5A864BCC-B490-5532-83AB-2E4109BB83C31 *EXPLOIT*
 45D138AD-BEC6-552A-91EA-8816914CA7F4 7.5 https://vulners.com/githubexploit/45D138AD-BEC6-552A-91EA-8816914CA7F4 *EXPLOIT*
 CVE-2023-45802 5.9 https://vulners.com/cve/CVE-2023-45802
 CVE-2022-37436 5.3 https://vulners.com/cve/CVE-2022-37436
 CVE-2022-28614 5.3 https://vulners.com/cve/CVE-2022-28614
 CVE-2022-28330 5.3 https://vulners.com/cve/CVE-2022-28330
```
service detection performed. Please report any incorrect results at https://nmap.org/submit/ .

```

The output listed the vulnerabilities it detected on each host.

- Bpf security tools

I use bpftrace which is a high-level language to create and use ebpf traces.
More specifically implementing bpftrace oneliner.

Tracking Command Execution

Use Case: Monitor and log every command executed on a system, which can help in forensic analysis in case of a security breach.

```
bpftrace -e 'tracepoint:syscalls:sys_enter_execve { printf("Command
executed: PID %d, Comm %s, Filename %s\n", pid, comm,
str(args->filename)); }'
```

Explanation:

- **tracepoint:syscalls**
: This tracepoint is triggered whenever a command is executed.
- **printf("Command executed: ...")**: Prints the process ID, command name, and the filename of the executable being run.

Here is the commands i used

```
asa@ASAttar-ASUS:/$ cd ~
asa@ASAttar-ASUS:~$ cd Desktop/
asa@ASAttar-ASUS:~/Desktop$ cat > new_file
cat << EOF > new_file1
a
new
home
EOF
more new_file1
^C
asa@ASAttar-ASUS:~/Desktop$ vi new_file
asa@ASAttar-ASUS:~/Desktop$ chmod +x new_file
asa@ASAttar-ASUS:~/Desktop$ ./new_file
./new_file: line 1: !#: command not found
a
new
home
asa@ASAttar-ASUS:~/Desktop$ vi new_file
asa@ASAttar-ASUS:~/Desktop$ ./new_file
a
new
home
asa@ASAttar-ASUS:~/Desktop$ which bash
/usr/bin/bash
```

Here is the output of trace

The screenshot shows two terminal windows. The left window displays the output of a bpftrace command, listing numerous system calls with their process ID (PID), command name, and filename. Several commands are highlighted with red boxes: 'Comm bash, Filename /usr/bin/cat', 'Comm sh, Filename /home/asa/.local/bin/x-terminal-emulator', 'Comm bash, Filename /usr/bin/bash', 'Comm bash, Filename ./new_file', 'Comm new_file, Filename /usr/bin/cat', and 'Comm new_file, Filename /usr/bin/more'. The right window shows a file editor session where the user is creating a file named 'new_file' with the command 'cat << EOF > new_file'. The file contains the text 'a', 'new', 'home', 'EOF', 'more new file', and ends with '^C'.

```
root@ASAttar-ASUS:~# bpftrace -e 'tracepoint:syscalls:sys_enter_execve { printf("Command executed: PID %d, Comm %s, Filename %s\n", pid, comm, str(args->filename)); }'
Attaching 1 probe...
Command executed: PID 233300, Comm bash, Filename /usr/bin/cat
Command executed: PID 233304, Comm (xtract-3), Filename /usr/libexec/tracker-extract-3
Command executed: PID 233366, Comm (spatcher), Filename /usr/lib/NetworkManager/nm-dispatcher
Command executed: PID 233370, Comm nm-dispatcher, Filename /etc/NetworkManager/dispatcher.d/01-ifupdown
Command executed: PID 233371, Comm nm-dispatcher, Filename /etc/NetworkManager/dispatcher.d/01-ifupdown
Command executed: PID 233388, Comm (xtract-3), Filename /usr/libexec/tracker-extract-3
Command executed: PID 233426, Comm (xtract-3), Filename /usr/libexec/tracker-extract-3
Command executed: PID 233437, Comm (xtract-3), Filename /usr/libexec/tracker-extract-3
Command executed: PID 233453, Comm bash, Filename /usr/bin/vi
Command executed: PID 233472, Comm gsd-media-keys, Filename /bin/sh
Command executed: PID 233472, Comm sh, Filename /home/asa/.local/bin/x-terminal-emulator
Command executed: PID 233472, Comm sh, Filename /usr/local/sbin/x-terminal-emulator
Command executed: PID 233472, Comm sh, Filename /usr/local/bin/x-terminal-emulator
Command executed: PID 233472, Comm sh, Filename /usr/sbin/x-terminal-emulator
Command executed: PID 233472, Comm sh, Filename /usr/bin/x-terminal-emulator
Command executed: PID 233472, Comm x-terminal-emul, Filename /home/asa/.local/bin/gnome-terminal
Command executed: PID 233472, Comm x-terminal-emul, Filename /usr/local/sbin/gnome-terminal
Command executed: PID 233472, Comm x-terminal-emul, Filename /usr/local/bin/gnome-terminal
Command executed: PID 233472, Comm x-terminal-emul, Filename /usr/sbin/gnome-terminal
Command executed: PID 233472, Comm x-terminal-emul, Filename /usr/bin/gnome-terminal
Command executed: PID 233475, Comm gnome-terminal, Filename /usr/bin/gnome-terminal.real
Command executed: PID 233479, Comm gnome-terminal-, Filename /bin/bash
Command executed: PID 233481, Comm bash, Filename /usr/bin/lesspipe
Command executed: PID 233482, Comm lesspipe, Filename /usr/bin/basename
Command executed: PID 233484, Comm lesspipe, Filename /usr/bin dirname
Command executed: PID 233485, Comm bash, Filename /usr/bin/dircolors
Command executed: PID 233486, Comm bash, Filename /usr/bin/which
Command executed: PID 233491, Comm (xtract-3), Filename /usr/libexec/tracker-extract-3
Command executed: PID 233514, Comm bash, Filename /usr/bin/chmod
Command executed: PID 233515, Comm bash, Filename ./new_file
Command executed: PID 233517, Comm bash, Filename /usr/bin/cat
Command executed: PID 233518, Comm bash, Filename /usr/bin/more
Command executed: PID 233531, Comm bash, Filename /usr/bin/vi
Command executed: PID 233539, Comm (xtract-3), Filename /usr/libexec/tracker-extract-3
Command executed: PID 233544, Comm bash, Filename ./new_file
Command executed: PID 233545, Comm new_file, Filename /usr/bin/cat
Command executed: PID 233546, Comm new_file, Filename /usr/bin/more
Command executed: PID 233605, Comm ThreadPoolForeg, Filename /usr/bin/google-chrome-stable
Command executed: PID 233606, Comm google-chrome-s, Filename /usr/bin/readlink
Command executed: PID 233607, Comm google-chrome-s, Filename /usr/bin dirname
Command executed: PID 233608, Comm google-chrome-s, Filename /usr/bin mkdir
Command executed: PID 233605, Comm google-chrome-s, Filename /opt/google/chrome/chrome
Command executed: PID 233609, Comm google-chrome-s, Filename /usr/bin/cat
Command executed: PID 233610, Comm google-chrome-s, Filename /usr/bin/cat
Command executed: PID 233754, Comm (xtract-3), Filename /usr/libexec/tracker-extract-3
Command executed: PID 233784, Comm (xtract-3), Filename /usr/libexec/tracker-extract-3
```

Here are some of the known commands i executed

Tracking opening files

Use Case : The script logs each time a file is opened on the system, useful for auditing and monitoring.

```
bpftrace -e 'tracepoint:syscalls:sys_enter_open { printf("%s %s\n", comm, str(args->filename)); }'
```

Explanation :

- **tracepoint:syscalls:** Triggered whenever a file is opened.
- **printf("%s %s\n", comm, str(args->filename)):** Prints the process name and filename being accessed.

```
rg ./local/lib/python3.10/site-packages/fontTools/designspaceLib  
rg ./local/lib/python3.10/site-packages/fontTools/cffLib  
rg ./local/lib/python3.10/site-packages/fontTools/cffLib/__pycach  
rg ./local/lib/python3.10/site-packages/fontTools/cu2qu/__pycache  
rg ./local/lib/python3.10/site-packages/fontTools/qu2cu  
rg ./local/lib/python3.10/site-packages/fontTools/subset  
rg ./local/lib/python3.10/site-packages/fontTools/colorLib  
rg ./local/lib/python3.10/site-packages/fontTools/colorLib/__pyca  
rg ./local/lib/python3.10/site-packages/fontTools/svgLib/__pycach  
rg ./local/lib/python3.10/site-packages/fontTools/varLib  
rg ./local/lib/python3.10/site-packages/fontTools/varLib/instance  
rg ./local/lib/python3.10/site-packages/fontTools/varLib/instance  
rg ./local/lib/python3.10/site-packages/matplotlib-3.9.0.dist-inf  
rg ./Desktop/Git/Operating_systems/Multi_threading_processing/curl  
rg ./Resume java  
rg ./Code local  
rg ./Code/Git  
rg ./config/Pinta  
rg ./config/dconf  
rg ./config/simple-scan  
rg ./config/nautilus  
rg ./Code/Git/bpftrace  
rg ./Code/Git/bpftrace/.gitignore  
rg ./Code/Git/bpftrace/.git/info/exclude  
rg ./Code/Git/bpftrace/tools  
rg ./Code/Git/bpftrace/tests/runtime/scripts  
rg ./Code/Git/bpftrace/tests/testlibs  
rg ./Code/Git/bpftrace/.github/codeql  
rg ./Code/Git/bpftrace/tests/codegen/llvm  
rg ./Code/Git/bpftrace/scripts  
rg ./Code/Git/bpftrace/cmake  
rg ./Code/Git/Cheat/Linux/Shell  
rg ./Code/Git/Cheat/Django/Http&Serializers  
rg ./config/texstudio  
rg ./config/texstudio/completion  
rg ./config/texstudio/completion/autogenerated  
rg ./Code/Git/Compiler_Design/convert2LL1  
rg ./Code/Git/Compiler_Design/Ex._Proj  
rg ./Code/Git/Compiler_Design/Ex._Proj/IL_Generator  
rg ./Code/Git/Compiler_Design/Ex._Proj/IL_Generator/.idea  
rg ./Code/Git/Compiler_Design/Ex._Proj/IL_Generator/.idea/.gitigno  
rg ./Code/Git/Compiler_Design/Ex._Proj/ILGenerator_project/ILGener  
rg ./Code/Git/Compiler_Design/Ex._Proj/ILGenerator_project/ILGener  
rg ./Code/Git/Compiler_Design/Ex._Proj/ILGenerator_project/ILGener  
rg ./Code/Git/Compiler_Design/Ex._Proj/ILGenerator_project/ILGener
```

Output

Syscall count by thread name

Use case : This **bpftrace** script counts the number of system calls made by each process, which can help in understanding the behavior and resource usage of applications.

```
bpftrace -e 'tracepoint:raw_syscalls:sys_enter { @[comm] = count(); }'
```

Explanation:

- **tracepoint:raw_syscalls**
: Triggered whenever any system call is made.
- **@[comm] = count();**: Aggregates and counts the number of system calls made by each process name (**comm**).

```
root@ASAttar-ASUS:~# sudo bpftrace -e 'tracepoint:raw_syscalls:sys_enter { @[comm] = count(); }' > /tmp/comm.log
Attaching 1 probe...
```

<pre>@[ibus-portal]: 2 @[Service Thread]: 2 @[xdg-document-po]: 2 @[fwupd]: 2 @[pool-/usr/libex]: 3 @[gnome-keyring-d]: 3 @[rsyslogd]: 3 @[boltd]: 3 @[Common-Cleaner]: 4 @[tracker-miner-f]: 4 @[kerneloops]: 4 @[packagekitd]: 5 @[wpa_supplicant]: 6 @[pool-udisksd]: 6 @[pipewire]: 6 @[cron]: 7 @[threaded-ml]: 7 @[gsd-sharing]: 8 @[Connection evic]: 10 @[pool-gnome-shel]: 11 @[BatteryStatusNo]: 12 @[udisksd]: 12 @[systemd-timesyn]: 15 @[systemd-resolve]: 15 @[CacheThread_Blo]: 17 @[GpuMemoryThread]: 19 @[GpuWatchdog]: 20 @[GUsbEventThread]: 26 @[MTP Download Se]: 26 @[rs:main Q:Reg]: 27 @[in:imuxsock]: 27 @[rtkit-daemon]: 28 @[C1 CompilerThre]: 40 @[C2 CompilerThre]: 40 @[alsa-sink-ALC29]: 42 @[inotify_reader]: 51 @[gvfs-afc-volume]: 52</pre>	<pre>@[QDBusConnection]: 64 @[update-notifier]: 64 @[hiddify]: 64 @[upowerd]: 66 @[pulseaudio]: 73 @[gsd-housekeepin]: 75 @[gsd-wacom]: 96 @[ibus-x11]: 96 @[gnome-terminal.]: 96 @[gsd-xsettings]: 96 @[gsd-keyboard]: 96 @[gsd-media-keys]: 97 @[evolution-alarm]: 99 @[gsd-power]: 100 @[cat]: 104 @[snap-store]: 112 @[NetworkManager]: 116 @[snapd-desktop-i]: 122 @[gsd-color]: 124 @[systemd]: 127 @[HangWatcher]: 132 @[ls]: 134 @[gnome-control-c]: 136 @[apache2]: 162 @[dockerd]: 164 @[ThreadPoolServ]: 194 @[ticker-schedule]: 196 @[VM Thread]: 198 @[G1 Service]: 199 @[snapd]: 204 @[MTP Main Sessio]: 224 @[io.flutter.ui]: 264 @[jemalloc_bg_thd]: 265 @[dbus-daemon]: 294 @[gnome-sh:gdrv0]: 303 @[systemd-journal]: 326 @[gjs]: 419</pre>
---	---

Read bytes by thread name

Use Case : This **bpftrace** script monitors and sums the total number of bytes read by each process, useful for analyzing data usage and identifying processes with high read activity.

```
bpftrace -e 'tracepoint:syscalls:sys_exit_read /args->ret/ { @[comm] = sum(args->ret); }'
```

Explanation:

- **tracepoint:syscalls**
: Triggered when a **read** system call exits.
- **/args->ret/**: Ensures only successful reads (where **args->ret** is non-zero) are counted.
- **@[comm] = sum(args->ret);**: Aggregates the total number of bytes read by each process, indexed by process name (**comm**).

```
root@ASAttar-ASUS:~# bpftrace -e 'tracepoint:syscalls:sys_exit_read /args->ret/ { @[comm] = sum(args->ret); }'
Attaching 1 probe...
^C

@[threaded-ml]: -86
@[dockerd]: 2
@[sudo]: 3
@[containerd]: 4
@[xtract-3]): 5
@[GpuWatchdog]: 5
@[gsd-print-notif]: 16
@[gsd-rfkill]: 16
@[gsd-screensaver]: 16
@[gsd-power]: 16
@[at-spi2-registr]: 16
@[rtkit-daemon]: 16
@[gsd-sharing]: 16
@[gsd-smartcard]: 16
@[gsd-wacom]: 16
@[gsd-xsettings]: 16
@[gsd-keyboard]: 16
@[boltd]: 16
@[gsd-a11y-settin]: 16
@[gsd-sound]: 16
@[gsd-datetime]: 16
@[gvfs-afc-volume]: 24
@[gvfs-ftp-volume]: 24
@[Monitor thread]: 24
@[gsd-color]: 24
@[gvfs-udisks2-vo]: 24
@[gvfs-gphoto2-vo]: 24
@[gvfs-goa-volume]: 24
@[xdg-document-po]: 40
@[ibus-portal]: 40
@[gnome-keyring-d]: 40
@[pipewire]: 40
@[ThreadPoolServ]: 56
@[xdg-desktop-por]: 56
@[pulseaudio]: 78
@[bash]: 99
@[pipewire-media-]: 105
@[Telegram]: 168
@[inotify_reader]: 192
@[thermal]: 204
@[gsd-media-keys]: 216
@[io.flutter.ui]: 240
@[tracker-miner-f]: 283
@[alsa-sink-ALC29]: 320
@[gnome-session-b]: 328
@[dbus-daemon]: 334
@[systemd-udevd]: 360
@[MTP Main Sessio]: 569
@[gvfsd]: 572
```

Read size distribution by thread name

Use Case : This `bpftrace` script captures and analyzes the distribution of read sizes for each process, useful for understanding how different processes use read operations.

```
bpftrace -e 'tracepoint:syscalls:sys_exit_read { @[comm] = hist(args->ret); }'
```

Explanation:

- **tracepoint:syscalls**
: Triggered when a `read` system call exits.
- **@[comm] = hist(args->ret);**: Creates a histogram of read sizes (`args->ret`) for each process (`comm`), showing how frequently different read sizes occur.

```
root@ASAttar-ASUS:~# bpftrace -e 'tracepoint:syscalls:sys_exit_read { @[comm] = hist(args->ret); }'
Attaching 1 probe...
^C

@[Telegram]:
[8, 16)           1 |████████████████████████████████████████████████████████████████|
[16, 32)          0 |
[32, 64)          0 |
[64, 128)         0 |
[128, 256)        0 |
[256, 512)        0 |
[512, 1K)          0 |
[1K, 2K)          1 |████████████████████████████████████████████████████████████████|
[2K, 4K)          0 |
[4K, 8K)          0 |
[8K, 16K)         0 |
[16K, 32K)        0 |
[32K, 64K)        0 |
[64K, 128K)       0 |

@[ibus-engine-sim]:
[8, 16)           2 |████████████████████████████████████████████████████████████████|
[16, 32)          0 |
[32, 64)          0 |
[64, 128)         0 |
[128, 256)        0 |
[256, 512)        0 |
[512, 1K)          0 |
[1K, 2K)          1 |████████████████████████████████████████████████████████████████|
[2K, 4K)          0 |
[4K, 8K)          0 |
[8K, 16K)         0 |
[16K, 32K)        0 |
[32K, 64K)        0 |
[64K, 128K)       0 |

@[io.flutter.ui]:
[8, 16)           2 |████████████████████████████████████████████████████████████████|
[16, 32)          0 |
[32, 64)          0 |
[64, 128)         0 |
[128, 256)        0 |
[256, 512)        0 |
[512, 1K)          0 |
[1K, 2K)          1 |████████████████████████████████████████████████████████████████|
[2K, 4K)          0 |
[4K, 8K)          0 |
[8K, 16K)         0 |
[16K, 32K)        0 |
[32K, 64K)        0 |
[64K, 128K)       0 |

@[sudo]:
[1]               1 |████████████████████████████████████████████████████████████████|
[2, 4)            1 |████████████████████████████████████████████████████████████████|
[4, 8)            0 |
[8, 16)           0 |
[16, 32)          0 |
[32, 64)          0 |
[64, 128)         0 |
[128, 256)        0 |
[256, 512)        0 |
[512, 1K)          0 |
[1K, 2K)          0 |
[2K, 4K)          0 |
[4K, 8K)          0 |
[8K, 16K)         0 |
[16K, 32K)        0 |
[32K, 64K)        0 |
[64K, 128K)       0 |

@[systemd-oomd]:
[8, 16)           1 |████████████████████████████████████████████████████████████████|
[16, 32)          0 |
[32, 64)          0 |
[64, 128)         0 |
[128, 256)        0 |
[256, 512)        0 |
[512, 1K)          0 |
[1K, 2K)          1 |████████████████████████████████████████████████████████████████|
[2K, 4K)          0 |
[4K, 8K)          0 |
[8K, 16K)         0 |
[16K, 32K)        0 |
[32K, 64K)        0 |
[64K, 128K)       0 |

@[DartWorker]:
(..., 0)          1 |████████████████████████████████████████████████████████████████|
[0]               0 |
[1]               1 |████████████████████████████████████████████████████████████████|
[2, 4)            0 |
[4, 8)            0 |
[8, 16)           0 |
[16, 32)          0 |
[32, 64)          1 |████████████████████████████████████████████████████████████████|
[64, 128)         0 |
[128, 256)        0 |
[256, 512)        0 |
[512, 1K)          0 |
[1K, 2K)          0 |
[2K, 4K)          0 |
[4K, 8K)          0 |
[8K, 16K)         0 |
[16K, 32K)        0 |
[32K, 64K)        0 |
[64K, 128K)       0 |

@[InputThread]:
(..., 0)          2 |████████████████████████████████████████████████████████████████|
[0]               0 |
[1]               0 |
[2, 4)            0 |
[4, 8)            0 |
[8, 16)           0 |
[16, 32)          0 |
[32, 64)          0 |
[64, 128)         2 |████████████████████████████████████████████████████████████████|
[128, 256)        0 |
[256, 512)        0 |
[512, 1K)          0 |
[1K, 2K)          0 |
[2K, 4K)          0 |
[4K, 8K)          0 |
[8K, 16K)         0 |
[16K, 32K)        0 |
[32K, 64K)        0 |
[64K, 128K)       0 |

@[Xorg]:
[1]               2 |████████████████████████████████████████████████████████████████|
[2, 4)            0 |
[4, 8)            0 |
[8, 16)           0 |
[16, 32)          0 |
[32, 64)          2 |████████████████████████████████████████████████████████████████|
[64, 128)         0 |
[128, 256)        0 |
[256, 512)        0 |
[512, 1K)          0 |
[1K, 2K)          0 |
[2K, 4K)          0 |
[4K, 8K)          0 |
[8K, 16K)         0 |
[16K, 32K)        0 |
[32K, 64K)        0 |
[64K, 128K)       0 |
```

Show per-second syscall rates

Use Case : This **bpftrace** script measures and displays the rate of system calls per second, which is useful for monitoring system activity and detecting abnormal patterns.

```
bpftrace -e 'tracepoint:raw_syscalls:sys_enter { @ = count(); }  
interval:s:1 { print(@); clear(@); }'
```

Explanation:

- **tracepoint:raw_syscalls**: Triggered every time any system call is made.
- **@ = count();**: Counts the total number of system calls.
- **interval:s:1 { print(@); clear(@); }**: Every second, prints the count of system calls and then clears the counter to start fresh for the next interval.

```
root@ASAttar-ASUS:~# bpftrace -e 'tracepoint:raw_syscalls:sys_enter { @ = count(); } interval:s:1 { print(@); clear(@); }'  
Attaching 2 probes...  
@: 36055  
@: 17368  
asa@A
```

```
2008*  
2009  sudo cat /etc/passwd  
2010  echo "test" > /tmp/testfile  
2011  cat new_file  
2012  cat new_file1  
2013  vi new_file  
2014  sudo cat /etc/passwd  
2015  vi new_file  
2016  cat new_file1  
2017  echo "test" > /tmp/testfile  
2018  history  
asa@ASAttar-ASUS:~/Desktop$
```

```
.....  
@: 36055  
@: 17368  
@: 4288  
@: 3586  
@: 2294  
@: 6052  
@: 23562  
@: 7374  
@: 4976  
@: 4912  
@: 3875  
@: 2108  
@: 4432  
@: 5328  
@: 4624  
@: 3021  
@: 4528  
@: 5764  
@: 2998  
@: 7205  
@: 2166  
@: 2242  
@: 5190  
@: 8312  
@: 9375  
@: 10001  
@: 2246  
@: 17027  
@: 15531  
@: 40378  
@: 3859  
@: 2600  
@: 2072  
@: 2463  
@: 3375  
@: 3843
```

These point the syscall rates increases because i executed multiple syscalls which each involve more than one syscalls.

Trace disk size by PID and thread name

Use Case : This **bpftrace** script monitors and logs disk I/O operations, showing the amount of data read or written by each process, useful for analyzing disk usage and performance.

```
bpftrace -e 'tracepoint:block:block_rq_issue { printf("%d %s %d\n", pid, comm, args->bytes); }'
```

Explanation:

- **tracepoint:block**
: Triggered when a disk I/O request is issued.
- **printf("%d %s %d\n", pid, comm, args->bytes)**: Prints the process ID (**pid**), process name (**comm**), and the number of bytes (**args->bytes**) involved in each disk I/O operation.

```
root@ASAttar-ASUS:~# bpftrace -e 'tracepoint:block:block_rq_issue { printf("%d %s %d\n", pid, comm, args->bytes); }'
Attaching 1 probe...
11561 ThreadPoolForeg 212992
433 jbd2/nvme0n1p12 40960
84 kworker/0:1H 0
```

```
Attaching 1 probe...
6712 ThreadPoolForeg 73728
660 jbd2/nvme0n1p10 16384
248 kworker/9:1H 0
248 kworker/9:1H 4096
6712 ThreadPoolForeg 4096
254 kworker/0:1H 0
6712 ThreadPoolForeg 4096
6712 ThreadPoolForeg 12288
6712 ThreadPoolForeg 4096
6712 ThreadPoolForeg 4096
6712 ThreadPoolForeg 4096
6712 ThreadPoolForeg 8192
6712 ThreadPoolForeg 4096
254 kworker/0:1H 0
660 jbd2/nvme0n1p10 24576
248 kworker/9:1H 0
```

Monitor Page Faults by Process

Use Case : This **bpftrace** script tracks the number of page faults occurring per process, useful for identifying processes with high memory access issues.

```
bpftrace -e 'software:faults:1 { @[comm] = count(); }'
```

Explanation:

- **software:faults:1**: Monitors page faults, which occur when a process accesses memory that is not currently mapped.
- **@[comm] = count();**: Counts and aggregates the number of page faults by process name (**comm**).

```
root@ASAttar-ASUS:~# bpftrace -e 'software:faults:1 { @[comm] = count(); }'
Attaching 1 probe...
^C

@[G1 Service]: 1
@[Monitor Deflati]: 1
@[gdbus]: 1
@[MTP Main Sessio]: 2
@[dockerd]: 2
@[alsa-sink-ALC29]: 2
@[DartWorker]: 2
@[gsd-color]: 2
@[dbus-daemon]: 3
@[ticker-schedule]: 4
@[VM Periodic Tas]: 6
@[HangWatcher]: 7
@[GpuMemoryThread]: 10
@[gjs]: 11
@[gnome-terminal-]: 14
@[ThreadPoolServ]: 20
@[systemd]: 21
@[VizCompositorTh]: 50
@[io.flutter.ui]: 53
@[Thread (pooled)]: 60
@[run-parts]: 63
@[sh]: 66
@[gnome-shell]: 68
@[Compositor]: 70
@[systemd-journal]: 145
@[cat]: 157
@[ThreadPoolSingl]: 207
@[Xorg]: 219
@[Chrome_IOThread]: 255
@[Telegram]: 307
@[cron]: 510
@[systemd-udevd]: 648
@[bash]: 1488
@[Chrome_ChildIOT]: 3139
@[elasticsearch[A]: 4055
@[ThreadPoolForeg]: 4133
@[ServiceWorker t]: 9037
@[bpftrace]: 12332
@[chrome]: 12808
@[sudo]: 17102
```

We can see that for example chrome has a lot more page faults than the average due to its high consumption of memory

Track LLC Cache Misses by Process

Use Case : This **bpftrace** script counts the number of Last-Level Cache (LLC) cache misses for each process, useful for identifying performance bottlenecks or unusual behavior.

```
bpftrace -e 'hardware:cache-misses:1000000 { @[comm, pid] = count(); }'
```

Explanation:

- **hardware:cache-misses:1000000:** Monitors LLC cache misses, which occur when data is not found in the cache.
- **@[comm, pid] = count();:** Aggregates and counts the number of cache misses by process name (**comm**) and process ID (**pid**).

```
root@ASAttar-ASUS:~# bpftrace -e 'hardware:cache-misses:1000000 { @[comm, pid] = count(); }'
Attaching 1 probe...
^C

@[kworker/2:0, 244833]: 1
@[swapper/5, 0]: 1
@[bash, 240640]: 1
@[kworker/4:0, 245056]: 1
@[apache2, 860]: 1
@[bpftrace, 246427]: 1
@[kworker/u17:1, 244774]: 1
@[swapper/1, 0]: 1
@[VM Periodic Tas, 2239]: 1
@[chrome, 11561]: 1
@[swapper/4, 0]: 1
@[Chrome_ChildIOT, 11561]: 1
@[swapper/0, 0]: 2
@[chrome, 11516]: 3
@[swapper/3, 0]: 3
@[VizCompositorTh, 11560]: 3
@[systemd-oomd, 577]: 3
@[swapper/2, 0]: 4
@[Compositor, 53543]: 5
@[gnome-terminal-, 38249]: 5
@[chrome, 11560]: 5
@[Xorg, 2481]: 8
@[ServiceWorker t, 246414]: 12
@[gnome-shell, 2804]: 14
@[chrome, 53543]: 19
```

Here we can see that chrome even has the highest cache misses again because of the same reason as the previous one.

