□ THE ANALYST'S BIBLE: TACTICAL FIELD MANUAL

The Definitive SOC Command Reference. No Fluff. Pure Syntax.

□ INDEX

---

1. □ WINDOWS: LIVE TRIAGE & FORENSICS

A. Network Connections (Netstat/PowerShell)

1. `netstat -ano` (Basic mapping of ports to PIDs)
2. `netstat -anob` (Requires Admin: Shows executable name)
3. `netstat -f` (Resolves FQDNs - useful for spotting weird domains)
4. `netstat -e -t 5` (Ethernet stats, refresh every 5s)
5. `Get-NetTCPConnection` (PowerShell object-based netstat)
6. `Get-NetTCPConnection -State Establish` (Show only established)
7. `Get-NetTCPConnection -RemoteAddress 192.168.1.0/24` (Filter by subnet)
8. `Get-NetUDPEndpoint` (List UDP listeners)
9. `Get-SmbConnection` (List active SMB shares mapped)
10. `Get-SmbSession` (List who is connected to YOUR shares)
11. `Get-SmbOpenFile` (List files currently open via SMB)
12. `Get-DnsClientCache` (View local DNS cache)

13. `Clear-DnsClientCache` (Flush DNS)
14. `ipconfig /displaydns` (CMD version of DNS cache)
15. `route print` (View routing table - look for VPN/Tunnel interfaces)
16. `arp -a` (View ARP cache - look for spoofing)
17. `Get-NetAdapter` (List physical/virtual interfaces)
18. `Get-NetFirewallRule -Enabled True` (List active FW rules)
19. `Get-NetFirewallRule -Direction Inbound -Action Allow` (Audit inbound allowances)
20. `Test-NetConnection -ComputerName 8.8.8.8 -Port 53` (Ping/Port check)

B. Process Inspection

21. `tasklist` (Basic list)
22. `tasklist /v` (Verbose: Shows User context - CRITICAL)
23. `tasklist /svc` (Shows Service hosting per PID)
24. `tasklist /m` (Shows DLLs loaded per PID - noisy but useful)
25. `Get-Process` (Basic PS list)
26. `Get-Process -IncludeUserName` (Needs Admin: Shows owners)
27. `Get-Process | Where-Object {$_.MainWindowTitle}` (Find visible apps)
28. `Get-Process | Sort-Object CPU -Descending | Select -First 10` (Top CPU hogs)
29. `wmic process list brief` (WMIC legacy list)
30. `wmic process get name,parentprocessid,processid` (Parent-Child mapping)
31. `wmic process where "name='cmd.exe'" get commandline` (Get CMD Args)
32. `Get-WmiObject Win32_Process | Select Name, CommandLine` (PS version of above)
33. `Get-CimInstance Win32_Process | Select Name, ParentProcessId` (Modern PS)
34. `query process` (Terminal Services view)
35. `handle.exe -a -u` (Sysinternals: Show open handles)
36. `listdlls.exe` (Sysinternals: Show loaded DLLs)
37. `procexp.exe` (Process Explorer GUI)
38. `procmon.exe` (Process Monitor GUI)

C. User & Group Enumeration

39. `whoami` (Current user)
40. `whoami /priv` (Check integrity/privileges)
41. `whoami /groups` (Check SIDs)
42. `whoami /all` (Full dump)
43. `net user` (List local users)
44. `net user administrator` (Check Admin account details)
45. `net user /domain` (List Domain Users - NOISY)
46. `net user <username> /domain` (Targeted domain query)
47. `net localgroup` (List local groups)
48. `net localgroup administrators` (Who is Local Admin?)

49. `net localgroup "Remote Desktop Users"` (Check RDP access)
50. `net group "Domain Admins" /domain` (The Crown Jewels)
51. `net group "Enterprise Admins" /domain` (Forest Admins)
52. `net accounts` (Password policy)
53. `Get-LocalUser` (PS Local Users)
54. `Get-LocalGroupMember -Group Administrators` (PS Admin check)
55. `Get-ADUser -Filter * -Properties LastLogonDate` (Active Directory Module)
56. `cmdkey /list` (List stored Windows Credentials - often used by attackers)
57. `vaultcmd /list` (List Credential Vaults)

D. Registry & Persistence (The "Run" Keys)

58. `Get-ItemProperty HKLM:\Software\Microsoft\Windows\CurrentVersion\Run` (System Run)
59. `Get-ItemProperty HKLM:\Software\Microsoft\Windows\CurrentVersion\RunOnce` (Run Once)
60. `Get-ItemProperty HKCU:\Software\Microsoft\Windows\CurrentVersion\Run` (User Run)
61. `Get-ItemProperty HKCU:\Software\Microsoft\Windows\CurrentVersion\RunOnce`
62. `Get-ItemProperty HKLM:\Software\WOW6432Node\Microsoft\Windows\CurrentVersion\Run` (32-bit apps)
63. `reg query HKLM\System\CurrentControlSet\Services` (List services in Reg)
64. `reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon"` (Check Shell/Userinit)
65. `reg query HKLM\System\CurrentControlSet\Control\Session Manager\KnownDlls` (DLL Hijacking)
66. `reg query HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist` (GUI execution history)
67. `reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options"` (IFEO Debugger keys)
68. `reg query HKCU\Environment` (User Environment Variables)
69. `reg query "HKLM\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections` (RDP Status)

E. File System Forensics

70. `dir /a /s /b *.exe` (Recursive search for EXEs)
71. `dir /ah` (Show hidden files)
72. `dir /ods` (Sort by date ascending)
73. `attrib -h -s -r *.*` (Unhide all files in cwd)
74. `Get-ChildItem -Force` (List hidden in PS)
75. `Get-ChildItem -Recurse -Include *.ps1` (Find scripts)
76. `Get-FileHash -Algorithm SHA256 .\suspicious.exe` (Calculate Hash)
77. `Get-Content .\file.txt -Wait` (Like Linux 'tail -f')
78. `Get-Content .\file.txt -Stream Zone.Identifier` (Read MotW - Download source)
79. `type C:\Windows\System32\drivers\etc\hosts` (Check Hosts file)

80. `forfiles /p C:\Windows\System32 /m *.exe /c "cmd /c echo @path"` (Iterate files)
81. `cipher /c \path\to\file` (Check encryption status)
82. `manage-bde -status` (BitLocker status)
83. `fsutil usn readjournal C:` (Read USN Journal - Advanced)

F. Event Logs (Get-WinEvent)

84. `Get-WinEvent -ListLog *` (List all logs)
85. `Get-WinEvent Security -MaxEvents 10` (Head of Security log)
86. `Get-WinEvent -FilterHashtable @{LogName='Security';ID=4624}` (Logon Success)
87. `Get-WinEvent -FilterHashtable @{LogName='Security';ID=4625}` (Logon Fail)
88. `Get-WinEvent -FilterHashtable @{LogName='Security';ID=4720}` (User Created)
89. `Get-WinEvent -FilterHashtable @{LogName='Security';ID=4726}` (User Deleted)
90. `Get-WinEvent -FilterHashtable @{LogName='Security';ID=4732}` (Group Member Add)
91. `Get-WinEvent -FilterHashtable @{LogName='Security';ID=1102}` (Log Clear)
92. `Get-WinEvent -FilterHashtable @{LogName='System';ID=7045}` (Service Install)
93. `Get-WinEvent -FilterHashtable @{LogName='System';ID=104}` (Log Clear System)
94. `Get-WinEvent -FilterHashtable @{LogName='Microsoft-Windows-PowerShell/Operational';ID=4104}` (Script Block)
95. `Get-WinEvent -FilterHashtable @{LogName='Microsoft-Windows-Sysmon/Operational';ID=1}` (Process Create)
96. `Get-WinEvent -FilterHashtable @{LogName='Microsoft-Windows-Sysmon/Operational';ID=3}` (Net Connect)
97. `Get-WinEvent -FilterHashtable @{LogName='Microsoft-Windows-Sysmon/Operational';ID=11}` (File Create)
98. `Get-WinEvent -FilterHashtable @{LogName='Microsoft-Windows-Sysmon/Operational';ID=13}` (Reg Set)
99. `Get-WinEvent -FilterHashtable @{LogName='Microsoft-Windows-Sysmon/Operational';ID=22}` (DNS Query)
100. `wevtutil qe Security /f:text /c:5` (CMD Query Events)

G. Scheduled Tasks (Deep Dive)

101. `Get-ScheduledTask` (List all)
102. `Get-ScheduledTask | Where State -eq 'Ready'`
103. `Get-ScheduledTask | Get-ScheduledTaskInfo | Select TaskName, LastRunTime`
104. `schtasks /query /xml` (Export to XML)
105. `schtasks /delete /tn "MaliciousTask"` (Delete)

H. WMIC (The Old Magic)

106. `wmic startup list full`
107. `wmic service list brief`

108. `wmic process call create "calc.exe"` (Lateral Movement technique)
109. `wmic useraccount list full`
110. `wmic group list full`
111. `wmic nicconfig get description,ipaddress`
112. `wmic qfe list` (List Patches/Updates)
113. `wmic logicaldisk get name,size,freespace`
114. `wmic product get name,version` (List installed software - Slow)
115. `wmic share get name,path`

I. PowerShell (The New Magic)

116. `Get-HotFix` (List Updates)
117. `Get-Service | Select-Object Name, StartType, Status`
118. `Get-Clipboard` (View clipboard contents)
119. `Get-History` (View PS Session History)
120. `(Get-PSReadlineOption).HistorySavePath` (Find history file)
121. `cat (Get-PSReadlineOption).HistorySavePath` (Read history file)
122. `Invoke-WebRequest "http://ifconfig.me/ip"` (External IP check)
123. `Get-MpComputerStatus` (Defender Status)
124. `Get-MpThreat` (Defender detected threats)
125. `Set-MpPreference -DisableRealtimeMonitoring $true` (The attackers move)

---

2. □ LINUX: HARDENING & HUNTING

J. System & Account Triage

126. `id` (Current user context)
127. `who` (Who is logged in)
128. `w` (Who is logged in + what are they doing)
129. `last` (Login history)
130. `lastb` (Failed login history - BTMP)
131. `cat /etc/passwd` (List users)
132. `grep -vE "nologin|false" /etc/passwd` (List humans/services with shells)
133. `cat /etc/shadow` (Hashes - Root only)
134. `cat /etc/group` (Groups)
135. `visudo -c` (Check sudoers integrity)
136. `awk -F: '($3 == "0") {print}' /etc/passwd` (Find UID 0 users)
137. `cat /root/.bash_history` (Root history)
138. `cat /home/*/.bash_history` (User history)
139. `history` (Current session history)
140. `uptime` (System uptime)

141. `uname -r` (Kernel version)
142. `hostnamectl` (Host details)
143. `env` (Environment variables)

K. Network & Sockets

144. `ss -tulpn` (TCP/UDP, Listening, Process, Numeric)
145. `ss -ant` (Show all TCP, numeric)
146. `netstat -antup` (Classic version)
147. `lsof -i` (List open network files)
148. `lsof -i :80` (Who is using port 80?)
149. `lsof -i TCP:22`
150. `nc -zv 127.0.0.1 22` (Netcat port check)
151. `ip addr` (Interfaces)
152. `ip route` (Routing table)
153. `arp -a` (ARP cache)
154. `cat /etc/resolv.conf` (DNS servers)
155. `cat /etc/hosts` (Local DNS)
156. `tcpdump -i eth0 -n` (Sniff traffic)
157. `iftop` (Bandwidth monitor)

L. Process & Services

158. `ps aux` (All processes)
159. `ps -ef --forest` (Process tree view)
160. `top -c` (Realtime resource with full commands)
161. `htop` (Colorful top)
162. `chkconfig --list` (Legacy services)
163. `service --status-all` (Init.d services)
164. `systemctl list-units --type=service` (Systemd services running)
165. `systemctl list-unit-files --state=enabled` (Enabled at boot)
166. `systemctl status sshd`
167. `strace -p <pid>` (Debug process calls)
168. `watch -n 1 "ps -ef | grep www-data"` (Monitor specific user)

M. File System Forensics

169. `ls -la` (List all, hidden)
170. `ls -R` (Recursive)
171. `ls -latr` (Sort by time reverse - see recent at bottom)
172. `find / -name ".*"` (Find hidden files)
173. `find / -mtime -1` (Modified in last 24h)
174. `find / -atime -1` (Accessed in last 24h)

175. `find / -ctime -1` (Created/Change in last 24h)
176. `find / -perm -4000` (Find SUID binaries)
177. `find / -perm -2000` (Find SGID binaries)
178. `find / -type f -size +100M` (Huge files)
179. `find / -user www-data` (Files owned by web user)
180. `grep -r "base64" /var/www/html` (Search webroot for encoding)
181. `grep -r "shell_exec" /var/www/html` (Search for webshells)
182. `stat file.txt` (Detailed timestamps)
183. `file file.exe` (Determine file type magic bytes)
184. `strings file.exe` (Extract text from binary)
185. `diff file1 file2` (Compare files)
186. `md5sum file`
187. `sha256sum file`
188. `chattr -i file` (Remove immutable flag)
189. `lsattr` (List attributes)
190. `du -Sh / | sort -rh | head -5` (Disk Usage)

N. Log Files (The Usual Suspects)

191. `/var/log/syslog` (Debian/Ubuntu general)
192. `/var/log/messages` (RHEL/CentOS general)
193. `/var/log/auth.log` (Auth attempts - GOLD)
194. `/var/log/secure` (Auth on RHEL)
195. `/var/log/kern.log` (Kernel errors)
196. `/var/log/dmesg` (Boot logs)
197. `/var/log/cron` (Cron job logs)
198. `/var/log/boot.log`
199. `/var/log/apache2/access.log` (Web access)
200. `/var/log/nginx/access.log`
201. `tail -f /var/log/auth.log` (Follow live)
202. `journalctl -xe` (Systemd logs)
203. `journalctl -u sshd` (SSH logs via systemd)
204. `journalctl --since "1 hour ago"`

O. Auditd (The Linux Sysmon)

205. `auditctl -l` (List rules)
206. `auditctl -s` (Status)
207. `ausearch -m USER_LOGIN`
208. `ausearch -m EXECVE`
209. `ausearch -m AVC` (AppArmor/SELinux denials)
210. `ausearch -f /etc/shadow` (Who touched shadow?)

211. `aureport -l` (Login summary)
212. `aureport -p` (Process summary)

---

3. □ NETWORK: PACKET ANALYSIS (NTA)

P. Tcpdump Filters (Capture)

213. `tcpdump -i eth0` (Default interface)
214. `tcpdump -i any` (All interfaces)
215. `tcpdump -n` (No DNS resolution)
216. `tcpdump -nn` (No port resolution)
217. `tcpdump -v` (Verbose)
218. `tcpdump -X` (Show Hex/ASCII payload)
219. `tcpdump -A` (Show ASCII payload - Good for HTTP)
220. `tcpdump host 1.2.3.4` (Filter by IP)
221. `tcpdump src 1.2.3.4` (Filter source)
222. `tcpdump dst 1.2.3.4` (Filter dest)
223. `tcpdump net 192.168.1.0/24` (Subnet)
224. `tcpdump port 80`
225. `tcpdump portrange 21-23`
226. `tcpdump not port 22` (Exclude SSH)
227. `tcpdump "tcp[tcpflags] & (tcp-syn) != 0"` (Capture SYN packets)
228. `tcpdump -w monitor.pcap` (Write to file)
229. `tcpdump -r monitor.pcap` (Read file)
230. `tcpdump -G 3600 -w ROTATE_%Y%m%d.pcap` (Rotate every hour)

Q. Tshark (CLI Wireshark)

231. `tshark -D` (List interfaces)
232. `tshark -r file.pcap` (Read)
233. `tshark -r file.pcap -Y "ip.addr == 1.2.3.4"` (Display filter)
234. `tshark -r file.pcap -Y "http"`
235. `tshark -r file.pcap -Y "dns"`
236. `tshark -r file.pcap -Y "smb"`
237. `tshark -r file.pcap -T fields -e ip.src -e ip.dst` (Extract fields)
238. `tshark -r file.pcap -T fields -e http.host` (Extract Host headers)
239. `tshark -r file.pcap -T fields -e dns.qry.name` (Extract DNS queries)
240. `tshark -r file.pcap -q -z conv,ip` (IP Conversations)
241. `tshark -r file.pcap -q -z io,phs` (Protocol Hierarchy)

242. `tshark -r file.pcap --export-objects http,./dump` (Rip files)

R. Zeek (Bro) One-Liners

243. `zeek -r traffic.pcap` (Generate logs)
244. `cat conn.log | zeek-cut id.orig_h id.resp_h id.resp_p service`
245. `cat dns.log | zeek-cut query`
246. `cat http.log | zeek-cut host uri`
247. `cat ssl.log | zeek-cut server_name`
248. `cat files.log | zeek-cut mime_type md5` (File extraction metadata)
249. `cat weird.log` (Protocol anomalies)

S. Nmap (Network Mapping)

250. `nmap 192.168.1.1` (Scan single)
251. `nmap -sn 192.168.1.0/24` (Ping sweep)
252. `nmap -p 80,443 192.168.1.0/24` (Target ports)
253. `nmap -p- 192.168.1.1` (All 65535 ports)
254. `nmap -sV 192.168.1.1` (Version detection)
255. `nmap -O 192.168.1.1` (OS detection)
256. `nmap -A 192.168.1.1` (Aggressive)
257. `nmap --script vulners 192.168.1.1` (Vuln scan)
258. `nmap --script smb-os-discovery` (SMB Enumeration)

---

4. □ SIEM OPERATIONS

T. Splunk (SPL)

259. `index=*` (Search all)
260. `sourcetype=xmlwineventlog`
261. `host="server01"`
262. `source="/var/log/syslog"`
263. `| head 10`
264. `| tail 10`
265. `| stats count`
266. `| stats count by src_ip`
267. `| stats count by src_ip, dest_ip`
268. `| stats distinct_count(user) as "Unique Users"`
269. `| timechart span=1h count`
270. `| top limit=20 src_ip`
271. `| rare user`

272. `| sort - count`
273. `| rename src_ip as "Source Address"`
274. `| fields - _raw` (Performance boost)
275. `| table _time, src_ip, user, action`
276. `| lookup threat_intel_ip ip AS src_ip OUTPUT malicious_confidence`
277. **Brute Force:** `index=win EventCode=4625 | stats count by src_ip | where count > 20`
278. **Pass Spray:** `index=win EventCode=4625 | stats dc(user) as distinct_users by src_ip | where distinct_users > 10`
279. **New Admin:** `index=win EventCode=4732 Group_Name="Administrators"`
280. **Log Clearing:** `index=win EventCode=1102`
281. **Process Crash:** `index=win EventCode=1000`
282. **Rare Parent:** `index=win EventCode=1 | stats count by ParentImage, Image | sort count`
283. **Encoded PS:** `index=win EventCode=1 CommandLine="*-emp*"`
284. **Data Exfil:** `index=fw bytes_out > 10000000 | stats sum(bytes_out) by src_ip`

U. Elastic (KQL/Lucene)

285. `event.code: 4624`
286. `winlog.event_data.LogonType: 3` (Network Logon)
287. `process.name: "cmd.exe"`
288. `user.name: "Administrator"`
289. `kibana.alert.severity: "high"`
290. `destination.ip: 10.0.*` (CIDR search)
291. `NOT process.name: "svchost.exe"`
292. `event.category: "process" AND event.type: "start"`
293. `file.extension: "exe" OR file.extension: "dll"`
294. `process.command_line: *mimikatz*`

---

5. □ ADVANCED: MEMORY & MALWARE

V. Volatility 3 (Memory Forensics)

295. `python3 vol.py -f mem.dmp windows.info`
296. `python3 vol.py -f mem.dmp windows.pslist`
297. `python3 vol.py -f mem.dmp windows.psscan` (Unlinked procs)
298. `python3 vol.py -f mem.dmp windows.pstree` (Parent/Child)
299. `python3 vol.py -f mem.dmp windows.procdump --pid <PID>` (Extract exe)
300. `python3 vol.py -f mem.dmp windows.dlllist`
301. `python3 vol.py -f mem.dmp windows.handles`
302. `python3 vol.py -f mem.dmp windows.netscan` (Connections)
303. `python3 vol.py -f mem.dmp windows.netstat`

304. `python3 vol.py -f mem.dmp windows.malfind` (Injected code)
305. `python3 vol.py -f mem.dmp windows.cmdline`
306. `python3 vol.py -f mem.dmp windows.registry.printkey`
307. `python3 vol.py -f mem.dmp windows.registry.hivelist`
308. `python3 vol.py -f mem.dmp windows.svcscan`
309. `python3 vol.py -f mem.dmp windows.driverscan`
310. `python3 vol.py -f mem.dmp windows.memmap` (Memory pages)

W. Malware Static Analysis

311. `strings malware.exe` (ASCII strings)
312. `strings -el malware.exe` (Wide/Unicode strings)
313. `floss malware.exe` (Obfuscated strings tool)
314. `pestudio malware.exe` (GUI Analysis)
315. `capa malware.exe` (Identify capabilities)
316. `peframe malware.exe`
317. `objdump -d malware.exe` (Disassemble)
318. `upx -d malware.exe` (Decompress UPX)
319. `binwalk malware.exe` (Find embedded files)
320. `foremost -i memory.dmp` (Carve files)

X. Base64 & Decoding

321. `echo "base64string" | base64 -d`
322. `certutil -decode input.txt output.bin` (Windows native)
323. `python3 -c "import base64; print(base64.b64decode('...'))"`
324. `xxd -r -p input.hex output.bin` (Reverse Hex)

---

6. □ INCIDENT RESPONSE: CONTAINMENT

325. `netsh advfirewall set allprofiles state on`
326. `netsh advfirewall set allprofiles firewallpolicy blockinbound,blockoutbound` (Hard Cut)
327. `ipconfig /release` (DHCP drop)
328. `Disable-NetAdapter -Name "Ethernet"` (Kill Interface)
329. `ufw enable`
330. `ufw default deny incoming`
331. `ufw default deny outgoing`
332. `iptables -P INPUT DROP`
333. `iptables -P OUTPUT DROP`

334. `ifconfig eth0 down`
335. `taskkill /PID 1234 /F` (Windows Force)
336. `taskkill /IM malware.exe /F` (By Name)
337. `kill -9 1234` (Linux SIGKILL)
338. `pkill -9 malware` (Pattern kill)
339. `net user <user> /active:no` (Windows Disable)
340. `usermod -L <user>` (Linux Lock)
341. `passwd -l <user>` (Linux Lock alt)

---

7. □ CONTAINER FORENSICS

Y. Docker Investigations

342. `docker ps` (List running)
343. `docker ps -a` (List all including stopped)
344. `docker inspect <container_id>` (View configuration/IPs)
345. `docker top <container_id>` (View processes inside)
346. `docker logs <container_id>` (View stdout logs)
347. `docker diff <container_id>` (View changed files)
348. `docker cp <container_id>:/path/to/file ./local` (Extract evidence)
349. `docker history <image_id>` (View image layers)
350. `docker network ls` (View networks)
351. `docker network inspect <network_id>`
352. `docker volume ls`
353. `docker volume inspect <volume_name>`
354. `docker exec -it <container> /bin/sh` (Enter shell)
355. `docker stats --no-stream` (Resource usage)
356. `docker events --since 24h` (Daemon events)
357. `docker export <container_id> > container.tar` (Snapshot FS)
358. `docker save <image_id> > image.tar` (Save image)

Z. Kubernetes (K8s) Triage

359. `kubectl get pods -A` (List all pods)
360. `kubectl get nodes -o wide`
361. `kubectl describe pod <pod_name>`
362. `kubectl logs <pod_name>`
363. `kubectl logs <pod_name> -c <container_name>` (Multi-container pod)
364. `kubectl logs -p <pod_name>` (Previous crashed instance logs)
365. `kubectl get events --sort-by='.lastTimestamp'`
366. `kubectl get secrets`

367. `kubectl get configmaps`
368. `kubectl get roles,rolebindings` (RBAC check)
369. `kubectl auth can-i create pods --as system:serviceaccount` (Priv check)
370. `kubectl exec -it <pod> -- /bin/sh`
371. `kubectl cp <pod>:/file ./local`
372. `kubectl get services`
373. `kubectl get ingress`
374. `kubectl cluster-info`
375. `kubectl api-resources --verbs=list`
376. `kubectl get networkpolicies`
377. `kubectl get serviceaccounts`
378. `kubectl get daemonsets`
379. `kubectl get deployments`
380. `kubectl get namespaces`

---

8. □ macOS FORENSICS

AA. System & Triage

381. `system_profiler SPSoftwareDataType` (OS Info)
382. `log show --predicate 'eventMessage contains "password"' --last 1h` (Unified Log)
383. `log show --style syslog --last 1d`
384. `kextstat | grep -v com.apple` (Non-Apple Kernel Extensions)
385. `launchctl list` (List LaunchDaemons/Agents - Persistence)
386. `ls -la /Library/LaunchDaemons`
387. `ls -la /Library/LaunchAgents`
388. `ls -la ~/Library/LaunchAgents`
389. `ls -la /Library/StartupItems` (Legacy persistence)
390. `pmset -g log` (Power management logs - sleep/wake)
391. `csrutil status` (SIP status)
392. `spctl --status` (Gatekeeper status)
393. `history` (Zsh history)
394. `lsof +c 0` (List open files)
395. `netstat -na | grep LISTEN`
396. `ps aux`
397. `dscacheutil -q group` (List groups)
398. `dscacheutil -q user` (List users)
399. `security dump-keychain` (Dump keychain info - risky)
400. `fdesetup status` (FileVault status)

AB. File System Artifacts

401. `/var/db/lsd/com.apple.lsd.map` (App usage)
402. `~/Library/Safari/History.db`
403. `~/Library/Messages/chat.db` (iMessage)
404. `mdfind -name "malware"` (Spotlight search CLI)
405. `mdls <file>` (Metadata list)
406. `xattr -l <file>` (Extended attributes - Quarantine tag)
407. `ls -lO` (List file flags like hidden)
408. `stat -x <file>`
409. `hdiutil info` (Mounted disk images)
410. `diskutil list`

---

9. □ ACTIVE DIRECTORY DEEP DIVE

AC. Enumeration (PowerView/Native)

411. `Get-NetDomain` (Domain info)
412. `Get-NetDomainController`
413. `Get-NetComputer -Ping` (Live computers)
414. `Get-NetGroupMember -GroupName "Domain Admins"`
415. `Get-NetShare`
416. `Get-NetGPO`
417. `Get-NetGPOGroup` (Restricted groups)
418. `Get-ObjectAcl -SamAccountName Administrator -ResolveGUIDs` (ACLs)
419. `Find-LocalAdminAccess` (Where am I admin?)
420. `Get-DomainTrust`
421. `Get-NetUser -SPN` (Kerberoasting targets)
422. `Get-DFSShare`
423. `nltest /domain_trusts`
424. `nltest /dclist:domain`
425. `dsquery user`
426. `dsquery group`
427. `dsquery computer`
428. `repadmin /replsummary` (Replication health)
429. `repadmin /showrepl`
430. `dcdiag` (DC Diagnostics)

AD. Attack Detection (Specifics)

431. `(Get-ADUser -Filter {AdminCount -eq 1}).SamAccountName` (Protected users)

432. `Get-ADUser -Filter {DoesNotRequirePreAuth -eq $true}` (AS-REP Roasting)
433. `Get-ADObject -Filter {msDS-AllowedToDelegateTo -ne "$null"}` (Constrained Delegation)
434. `Get-ADUser -Filter {SidHistory -ne "$null"}` (SID History injection)
435. `Get-ADGroup -Filter {Name -like "*Admin*"}` (Shadow Admin groups)
436. `Get-WinEvent -LogName "Directory Service"`
437. `ntdsutil "ac i ntds" "quit" "quit"` (Audit snapshot usage)
438. `klist` (List Kerberos tickets)
439. `klist purge` (Clear tickets)
440. `klist tgt`

---

10. □ REVERSE ENGINEERING & DEBUGGING

AE. GDB (Linux Debugger)

441. `gdb ./program`
442. `run <args>`
443. `break main`
444. `info registers`
445. `x/10i $rip` (Examine instructions at IP)
446. `x/10s $rsp` (Examine string at SP)
447. `next` / `step`
448. `continue`
449. `disassemble main`
450. `bt` (Backtrace)

AF. Radare2 (r2)

451. `r2 ./program`
452. `aaa` (Analyze all)
453. `afl` (Analyze functions list)
454. `pdf @ main` (Print Disassembly Function)
455. `iz` (Print strings in data section)
456. `ii` (Print imports)
457. `s main` (Seek to main)
458. `V` (Visual mode)
459. `VV` (Visual Graph mode)
460. `wx 9090` (Write NOPs)

AG. Windows Debugging (WinDbg)

461. `lm` (List modules)
462. `!analyze -v` (Auto analyze crash)
463. `kb` (Stack trace)
464. `da <address>` (Dump ASCII)
465. `du <address>` (Dump Unicode)
466. `dd <address>` (Dump Dword)
467. `!process 0 0` (List all procs)
468. `!process <addr> 7` (Detail proc)
469. `!gle` (Get Last Error)
470. `bp <address>` (Breakpoint)

---

11. □ YARA & SIGMA ENGINEERING

AH. YARA Rule Syntax

471. `rule Detect_Malware { ... }`
472. `strings: $a = "evil_string"`
473. `strings: $b = { 4D 5A 90 00 }` (Hex/Magic Bytes)
474. `condition: $a and $b`
475. `condition: any of them`
476. `condition: $a at 0` (String at entry)
477. `condition: filesize < 100KB`
478. `condition: uint16(0) == 0x5A4D` (MZ Header)
479. `yara rule.yar file.exe` (Run scan)
480. `yara -r rule.yar directory/` (Recursive scan)

AI. Sigma Rule Basics

481. `title: Suspicious Process`
482. `logsource: category: process_creation`
483. `detection: selection: Image: '*\cmd.exe'`
484. `detection: condition: selection`
485. `dict_to_sigma.py` (Custom tool)
486. `sigma-cli check rule.yml`
487. `sigma-cli convert -t splunk rule.yml`
488. `sigma-cli convert -t elasticsearch rule.yml`

AJ. Miscellaneous & Google Hacking (Dorks)

489. `site:pastebin.com "password"`
490. `filetype:config "db_password"`

491. `inurl:gitlab "password"`
492. `inurl:s3.amazonaws.com "secret"`
493. `ext:log "username"`
494. `intitle:"index of" "backup"`
495. `intext:"BEGIN RSA PRIVATE KEY"`
496. `ssh-keygen -t ed25519` (Generate secure keys)
497. `openssl s_client -connect google.com:443` (Check SSL)
498. `openssl x509 -in cert.pem -text -noout` (Read Cert)
499. `gpg --gen-key`
500. `base64 /dev/urandom | head -c 32` (Generate Random Pass)
