|  |
| --- |
| **Analyzing Network Evidence**: *Overview* (Attacker’s reconnaissance and scanning behavior: CSIRT may identify extra infra attempting to compromise perimeter systems; Initial infection: Multi-stage exploits/malware, must log outbound traffic to identify C2; Lateral movement: NetFlow logs provide this data, look for unusual patterns, log success & denied attempts; C2: logs + pcap + NetFlow, look for remote procedure calls, RDP, VPN logins from outside n/w, log success + denied attempts; Data exfil: proxy logs has URL, target IP, NetFlow for internal to external, pcaps for exfil data, src, dst) | *Firewall and proxy logs* (Manual log review: Low cost, limited data only, web app data on a day; Filtered log review: tools can filter by parameters, easy to prioritize efforts, doesn’t quickly indicate malicious activity; Log file searching: regex/Boolean, limit to a time period, src ip, or condition; Log file correlation: part of SIEM with rulesets, powerful, automated, upfront labor to configure/tune to env, AI/ML can help but needs data, has bias, and false positives when normal changes; Log file data mining: extract meaning, sentiment analysis/interpretive is hard, use threat intel, no actionable intel, insights into malware, ELK; *DNS blacklists python script* (from bitbucket) to identify bad IP addr/domains; explicit deny approach); SIEM tools: centralized collection, aggregation, analysis of logs, cross-component/cross-system visibility, series of events across systems indicate the issue| *NetFlow* (src, dst, sport, dport, proto, packets, bytes, flows; **common ports**:1:tcpmux, 5:rje, 7:echo, 9:discard, 11:systat, 13:daytime, 17:qotd, 18:msp, 19:chargen, 20:ftp-data, 21:ftp, 22:ssh, 23:telnet, 25:smtp, 37:time, 39:rlp, 42:nameserver) | *Packet captures* (src ip, ds tip, domains, ports, content; UsefulTools: mergecap, editcap; Pcap tools: Moloch, [**Wireshark**: Time: correlate by date/time | Name Resolution: toggle b/w IP addr and domains of src & dst| Display Filters: by services and ports | colorize packets | protocol identification | hostname identification | packet stream examination (TTPs)]) |
| **Analyzing System Memory**: **RAM contains**: Running processes, Loaded DLLs, Loaded device drivers, Open registry keys, N/w connections, Cmd history | Memory analysis methodology: Consistent, legally defensible, takes less time to analyze (muscle memory), Stay in scope of search, Less error prone, Std of reports makes reports more believable/reliable, Reduces human error, Muscle memory of CSIRT, Trend analysis (same approach eliminates variables); **SANS six-part methodology**: 1. Identify rogue processes [misspelled processes names, Attempts to execute from illegitimate source], 2. Analyze process DLLs & handles [Malware coders insert malicious DLLs, aka **DLL Injection**: malicious DLL in process, point to external host, no overriding existing DLL], 3. Review n/w s [malware beacons to C2 servers, find IP addr and dig into it], 4. Look for evidence of **code injection** [**process hollowing**: add malicious code to suspended process, then activating it; unmapped sections used; need special tools for evidence], 5. Check for rootkit [achieving persistence via rootkit by hiding from OS is attacker’s goal; continuous/elevated access to system, undetected], 6. Dump suspicious process & drivers [save to n/w folder or removable media]; **Network connections methodology**: Suspicious n/w conn, Process name, Parent process ID, Entities | Memory analysis w/ **Redline**: Fireye (Mandiant) tool, Windows, use IoC to analyze; creates analysis session, process (PID) analysis | Memory analysis w/ **Volatility**: Python script, plugins to analyze, any OS, process list/scan/tree, DLL list, Handles plugin, LDR modules, xview, n/w analysis, mem dump| Memory analysis w/ **Strings** (*apt install binutils*, grep for IP addr or HTTP data). |
| **Analyzing System Storage**: Forensic platforms (**Qs to ask**: platform tested?, used successfully in criminal/civil proceedings?, how tool fits into IR plan? **Tool features**: file structure view [quickly review files w/ known locations], hex viewer [for viewing malware or custom exploits], web artifacts [social engg attacks, users navigated to malicious sites], email carving [malicious employees involved in illegal activities/policy violations], image viewer [law enf uses for child exploitation on system], metadata [files date/time, hash, location]); **Commercial tools**: OpenText EnCase (used in BTK Killer, criminal investigations, trained analyst), AccessData FTK (disk forensics, FBI uses), X-Ways Forensics (low cost, less resource intensive, run from USB)| Autopsy (GUI, SleuthKit toolset, timeline analysis, keyword search, web and email artifacts (malicious insider upload confidential docs; Social engg by C2 redirects to steal creds/download malicious code), filter results on known bad file hashes, attached devices, file metadata, deleted files, **ease of use**) | Master File Table (MFT) analysis (list of all files on system; export using Autopsy: Python script: ***analyzeMFT*** will product excel w/ date & time) | Registry analysis (fav target of malware coders; HKEY\_CURRENT\_USER, HKEY\_USERS, HKEY\_CLASSES\_ROOT, HKEY\_LOCAL\_MACHINE (HKLM: SAM, Security, Software, System), HKEY\_CURRENT\_CONFIG) |
| **Analyzing Log Files & SIEM**: Best practices for logging: Automation is helpful; Focus on high value/data systems; identify parts of dataset for security reviews; Tier storage (cloud storage for some + pair w/ cloud native data analytics tools; SIEM to help with triage (need sense of tuning) | Log mgmt.: **Establish logging as normal business practice**: (users have reasonable expectation of privacy absent from expressly stated monitor policy; logs enabled strictly for user’s malicious activity have legal issues; log policy must establish logging as normal business activity and users don’t have privacy expectation) | **Logging close to the event**: (logs not created close to event lose value as evidence in court, except automated logging) | **Knowledgeable person**: (context matters: who created and whether knowledgeable or not; logging software is better thru std schemas & ontological structure) | **Comprehensive logging**: (for as much enterprise as possible; consistent to support trend analysis; random logs has less value in court; ***Enclaving*** is bad as it limits visibility of logs; Plus it’s about security vs. transparency.) | **Qualified custodian**: (log policy naming a data custodian; person to speak for log procedure & software used to testify accuracy in court; *Log Custodian doesn’t exist in orgs* due to lack of awareness/priority, complexity of resp; resource constraints; no regulation pressure; assumed low risk) | **Document log system failures**: (too much fail can diminish value in court; fail must be documented w/ reason) | **Log file discovery**: (Made available to opposing legal counsel) | **Logs from compromised systems**: (Logs on those are suspect; use at risk) | **Original copies are preferred**: (log copied from source to storage media; archived off the system; establish chain of custody for log files) | **Heterogeneous enterprises**: Many log sources & single source w/ multiple logs | complexity is exponential | increased false positive & | can use AI/ML | never 100% compliance | no common values, so difficult to link events | log sources only record key information | log sources may have diff formats (e.g. date dd/mm/yyyy or mm-dd-yyyy OR comma or tab separated OR **SNMP / XML / binary** formats), log sources refer to its internal system clock | malware developers set explicit timezones to confuse log analysis | some logs for humans to read, and some in proprietary format | **SIEM**: aggregates log and event data from network sources, combine into single location. Allows CSIRT to observe activity across n/w, w/o checking each system | **SIEM functions**: Log aggregation: (central location; holistic visibility across all devices aggregated) | Log retention: (archive logs in orderly fashion; allows retrieval for compliance purposes) | Routine analysis: (periodic review of info; dashboards w/ # of conn, data flows, critical alerts; ad-hoc reporting for informing stakeholders/pull basis) | Alerting: (for specific conditions of malicious activity (from AV, IDS, IPS) | Event correlation: (link between events; helps in lateral movements or priv escalations attempts) | Incident response: **SOAR**: CSIRT queries, download of logs when needed for offline; reduces time for searches and event collection. | **SIEM tools**: Qradar, Splunk, LogRhythm, Elastic, Security Onion | **Windows event logs**: Security logs: (logons & logoff (from local or remote, *too many failed indicate brute-force or stolen creds*), group membership, program execution) | App logs: (used in process audit/troubleshooting; developers decide what to log) | System logs: (troubleshoot non-malicious activity; Windows OS creates these) | **Windows logs for Incident Responders**: 4624/4634: logon & logoff | 4625: acct failed logon | 4672: special priv assigned to new logon (admin level)| 4688: new process created (everytime process is run, **abused programs: PsExec, CMD.exe, Whoami.exe**)| 4748 & 4773: Kerberos service (TGT for elevated priv: **Kerberoasting**; Provides attackers w/ valid creds) | 5140: n/w share object was accessed (logged upon first access to n/w share) | 7045: new service installed (strains of malware will install itself as service) |
| **Writing the Incident Report**: CISA, DoD (Defense Industrial Base); **What to document**: 5 Ws and H (Who [is involved], When [date/time; std TZ; daylight or not], Where [location e.g office], What [action perform, e.g acquire memory], Why [reason for action performed], How [used playbooks or SOP; any deviations from SOP must be included]); Asset data, Event data, Timelines, Any record about event, Steps taken; Types of documentation: **Trouble** **Ticketing system:** [Track ticketing system outages, start+stop date/time, original reporting person, action performed, some area for notes; Major drawbacks: Verbose OUTPUT and often unstructured data]; **IR orchestration platforms**: e.g. FIR (Fast Incident Response): Analysts can i/p data, attach evidence files, collaborate with team members; *Advantage*: Automate info capturing w/ date/time, actions | Limit authz to groups | Track actions taken + info obtained; **Written reports**: Some incidents require extensive written report: *3 types of written reports*: **Executive summary**: (1-2 page, high-level bullet points of incident to sr. mgmt., brief synopsis of events, root cause, remediation recommendations) | **Incident report**: (detailed report; lot of readers in org; details of investigation, detailed root cause, through recommendations on preventing incident from reoccurring) | **Forensic report**: (Most detailed report; when forensics is done on logs, memory, disk images; length; technical; output from tools) | Sources of information for reports: personal observations: (users & analysts view of data on case) | Applications: (log files) | Network/host devices: (systems output reports be included) | Forensics tools: (automated reporting functions; file hashes)  Audience of reports: Executives (high-profile incidents, media involved); IT personnel (interested in what analysts found); Legal (find gaps in security or relevant procedures); Marketing (craft message to customers on external breach), Regulators (liability check w/ healthcare & financial institutions); Law enforcement (need copies of incident/forensics reports); Outside support (reports bring outside support up to speed quickly) | Incident report contents: Background, Events timeline, N/w infrastructure overview, Forensic analysis overview, Containment actions, Findings/Root cause analysis, Remediation, Final recommendations, Definitions | Forensic report contents: Examiner bio/background, Tools utilized, Evidence items, Forensics analysis, Tool output (e.g. Autopsy), Conclusions, Definitions, Exhibits. |
| **Malware Analysis for Incident Response**: Malware classifications (**Virus**: (intentionally malicious; attaches itself to clean files & spreads) | **Worms**: (self-replicating, spreads n/w w/o files) | **Trojans**: (user opens file; uses social engg; disguised as legitimate s/w, creates backdoors) | **Ransomware**: (encrypts files, demands pay) | **keylogger**: (hides in background of running system; for creds) | **Rootkits**: (Conceal malicious code like **Remote Access Trojan (RAT)**; Remote command of injected system; Admin control over system w/o detection to attackers) | **Information-stealing malware**: coded for single purpose; capture credit cards, creds; *shylock malware* for banking logins); | **Backdoor**: variation of remote access; infects system, allows attacker take control of infected system | **Downloader**: *multi-stage malware* program; first infects, then reaches out to remote server for more code; for bypassing security controls | **Botnet**: series of computers, controlled by central system on internet; DDoS attack | **File wipers**: destroys files or infect **Master Boot Record (MBR)** by modifying records to make files inaccessible) | Malware analysis overview (or **malware reverse engg** | **Static Analysis**: Examining without executing code; uses fingerprinting [most basic, code hash], AV scanning [multiple AV vendors or virustotal], File format [compilation time, functions, strings, menus, icons or **PE-format** apps], packer analysis [to check for bypass AV code, malware code uses compression or encryption, no file hash left, difficult by static analysis], Disassembly [reversing by software, see assembly code, identify malware actions]; Advantages of static analysis: safe, can’t infect; Disadvantages: slow, expensive (disassembly, packer analysis), limited visibility, can’t find zero-day exploits, only signature-based eval; Static analysis tools: ClamAV, PeStudio, YARA, REMnux | **Dynamic Analysis**: Run malware in controlled env, observe behavior, compare before and after execution of system, aka **defined point analysis**; **Runtime behavior analysis**: Tools like process explorer to observe behavior of malware when running; Advantages of dynamic analysis: Fun, see effects of malware in realtime, insights into IoC, can find zero-day exploits; Disadvantages of dynamic: need sandbox & resources to build it, risk of propagation/may infect env, need env that matches production; Dynamic analysis tools: Cuckoo Sandbox; Memory analysis with Volatility, Process Explorer (no cost/CPU usage), Process Spawn Controller script (PowerShell)) |
| **Threat Intelligence**: **Evidence-based**: obtained thru good evidence collection methods (e.g. malware analysis); analysts rely on it for validity; **Utility**: clarity on context, data, behavior/methods to evaluate incident against others in similar nature; **Actionable**: key element that separates data/info from threat intel; can be specific seq of events or focus area of incident; security control is implement or not; what cyberthreats org likely to face; **Adversary motivations**: Money, political goals, Disrupt competition/target, Ambition/Street cred/Notoriety/Chaos/Boredom, Power, Leverage over orgs/people: **MICE (Money, Ideology, Coercion, Ego); Threat actors:** Cybercriminals/**eCrime** (Targeted, Hacktivist, Unattributed), Hacktivism, Cyber espionage; **Social Engg techniques**: Eavesdropping (intercepting unauthz comm; Digital [**Evil Twin or MITM**] or Social [Overhear]), Shoulder surfing (in airplane), Dumpster diving (collect sensitive info; docs not shredded); *Repeat Offenders*; **Attack trends**: 41% of malware were new; 70% belong to 1 of 5 frequent attackers, using open-source tools; 23% publicly available malware; mostly for priv esc or lateral movement. | **RaaS (Ransomware as a service)**: PINCHY SPIDER, split in revenues/profits | **Malware as a Service (MaaS)**: VENOM SPIDER, developer on SKID (JS backdoor called More-Eggs), VenomKit (exploit for delivery of executable, DLL or PowerShell payload w/ MS word file as decoy; Include text/images as social engineering), Taurus Loader (malicious VBS MS word docs for JS w/ DLLs; Dev Steaker module, TeamViewer module, Recon module, Ransomware module) | **Access brokers**: gain backend access to multiple orgs/gov, sell access to criminals | **APT** (ninjas, not tanks): offensive cyber ops that uses continuous clandestine, sophisticated technique to gain access to system and remain inside for long; **Five stages of APT**: Gain Access (via infected file, junk email or app vuln) | Establish a Foothold (implant malware; create backdoor; code to cover tracks) | Deepen Access (password cracking; get admin rights) | Move Laterally (Access other systems) | Look, Learn, & Remain (Harvest info; keep process running indefinitely/withdraw once goal done; leave backdoor for future use) | Threat intelligence sources: *Internal, External* (Mandiant, Crowdstrike, Cisco, Fireeye, Palo Alto Unit 42)*, OSINT* | Twitter/social media, Adversarial TTPs (MITRE ATT&CK, CISA warning, ENISA warning), Commercial reports, commercial feeds(Talos, Falcon overwatch, unit 42, intel 471), Attribution info, Technical intel (ip, blacklists, FQDN/domains) | Threat intelligence platforms: **MISP** (**Malware Info Sharing Platform**) uses OVA (Open Virtualization Format)| Using threat intelligence (**Proactive**/Passive threat intel: CSIRT/SOC can fed into SIEM, CTI feeds, OSINT feeds, **membership in ISAC**, no zero-day exploit visibility; not up-to-date always w/ technology/trends; **Reactive**/Active threat intel: Analysts know during investigation of IoC without evidence, **Participation in ISAC**; IoC shared with target; Exposes intent/capability/technology and active response method exposed); **Types of Threat Intel**: **Tactical threat intel**: IOCs (e.g. C2 IP or MD5 hash of malware); IOAs (artifact on system; doesn’t indicate compromise); TTPs | **Operational Threat Intel**: specific attacks/events; real-time feeds/alerts; info on attacks & threat actors on larger industry | **Strategic threat intel**: High-level info on trends in attacks, patterns, risks, big-picture, what industries are target; change in technology | **Technical Threat Intelligence**: (too detailed on attacks like IP addr, malware sig, domain names used; helps on detect & block threats); **Pyramid of pain**: Hashes (Trivial to bypass), IP addr (easy), Domain names (Simple), N/w & host artifacts (Annoying), Tools (challenging), TTPs (Tough) | **Threat intel methodology**: *Direction* (identify users by threat based on need), *Collection* (data & info from its sources: gov agencies, 3rd party org, OSINT), *Processing* (relevance & reliability of data), *Analysis* (evaluate & combine other sources data), *Dissemination* (new intel sent to users for use) | **Cyber Kill Chain**: Recon -> Weaponization -> Delivery -> Exploit -> Install -> C2 -> Action on objective. |
| **Threats Hunting**: Process-driven exercise | **threat hunting maturity model**: HM0: Initial, HM1:Minimal, HM2:Procedural, HM3:Innovative, HM4:Leading | [**Threat hunting cycle**: Initiating event (trigger from CISA/FBI warning, new TTP release, new threat intel, new IOCs), Create a working hypothesis (focused endeavour; craft hypothesis for specific target), Leverage threat intel (focus on TTPs, intel reports, OSINT useful), Apply forensic techniques (test the hypothesis, examine proxy logs for suspicious URLs, DNS logs for resolve URLs to connect, Firewall logs for conn made), Identify new indicators (feedback loop, updated threat intel with our findings: unusual outbound traffic, anomalies in priv user accts, geographical anomalies, excessive login failures, excessive DB read volume, HTML response sizes, Excessive file requests, port-app mismatch, suspicious registry or system file changes, DNS req anomalies), Enrich the existing hypothesis (modify existing threat hunt hypothesis, new IOC found becomes new initiating event) ] | **MITRE ATT&CK**: **TA0001: Initial access**: adversary trying to get into n/w; **TA0002: Execution**: run malicious code; **TA0003:Persistence**: maintain foothold; **TA0004: Priv Esc**: gain high priv; **TA0005:Defense Evasion**: Avoid being detected; **TA0006:Credential Access**: steal account names/pwds; **TA0007:Discovery**: figure out env; **TA0008:Lateral movement**: move thru env; **TA0009:Collection**: gather data of interest for goal; **TA0010:Exfiltration**: steal data; **TA0040:Impact**: manipulate, interrupt, destroy systems/data | **Threat hunt planning**: Hypothesis, MITRE ATT&CK tactics, Threat intel, Evidence sources, Tools, Scope, Timeframe | **Threat hunt reporting**: Executive summary, Threat hunt plan, Forensics report, Findings, Recommendations. |

**Exploits not identified by monitoring for AV signature/patching systems**: Zero-day attack, Encrypted payloads embedded in non-lethal code, open-source malware (Petya > WannaCry > NotPetya), stolen NSA tools (shadow brokers), Stolen creds make it look like normal users.

**What We Can Learn from Analysis – Types of Indications**: Suspicious files/hashes | Unusual n/w traffic | Unauthz access attempts | System config changes | Perf degrade | Malware artifacts present

**Mask your activity**: VPN at entry point, Incognito mode, privacy overlay (ddg.go), privacy browser (brave, ToR), Use of VMs to switch across multiple OS/Browsers

**Average breakout time**: time from initial entry point to lateral movement (<2 hrs on average; 30% os cases in <30 min) | **Average Dwell time**: from first system breach to being detected (24 days average; Extreme cases like SolarWinds, it took over 12 months)