SOC Playbook: Hollow Process Behavior Detection (T1055.012)

1. Objective

Detect and respond to **process hollowing**, where a legitimate process is started in a suspended state, its memory is unmapped and replaced with malicious code, and then resumed to evade detection.

2. Scope

- Detect hollowing behavior across Windows systems.
- Identify use of APIs like CreateProcess (suspended), ZwUnmapViewOfSection, WriteProcessMemory, and ResumeThread.
- Track suspicious parent-child process relationships and memory changes.
- Enable rapid response to prevent malware execution and lateral movement.

3. Log Sources

Platform	Log Source	Description
Windows	Windows	Windows
Sysmon (Event IDs 1, 8, 10)	Sysmon (Event IDs 1, 8, 10)	Sysmon (Event IDs 1, 8, 10)
Process creation, remote thread, image load	Process creation, remote thread, image load	Process creation, remote thread, image load
Windows	Windows	Windows

4. Detection Rules / Alerts

Alert Name	Description	Triggers / Examples
Suspicious Suspended Process Start	A known process (e.g., svchost.exe, notepad.exe) started in suspended state	Uses CreateProcess(..., CREATE_SUSPENDED)
Unmap + Write to Remote Process	Use of ZwUnmapViewOfSection followed by WriteProcessMemory	Observed in Sysmon + EDR
Memory Injection + Resume	WriteProcessMemory followed by ResumeThread	Full hollowing sequence
Inconsistent Process Image and Memory	Executable loaded doesn't match command-line or binary path	Check loaded modules vs. disk path
Hollowing Known Binaries	Processes like svchost.exe, explorer.exe, or regsvr32.exe show abnormal behavior	Rare for these to be launched manually

High-Entropy Executable Memory	Suspicious executable memory sections in legit processes	No matching image file path, entropy > 7.5
Child Process with No Command Line	Legit process spawned with missing or blank command line	Can indicate spoofed or injected process

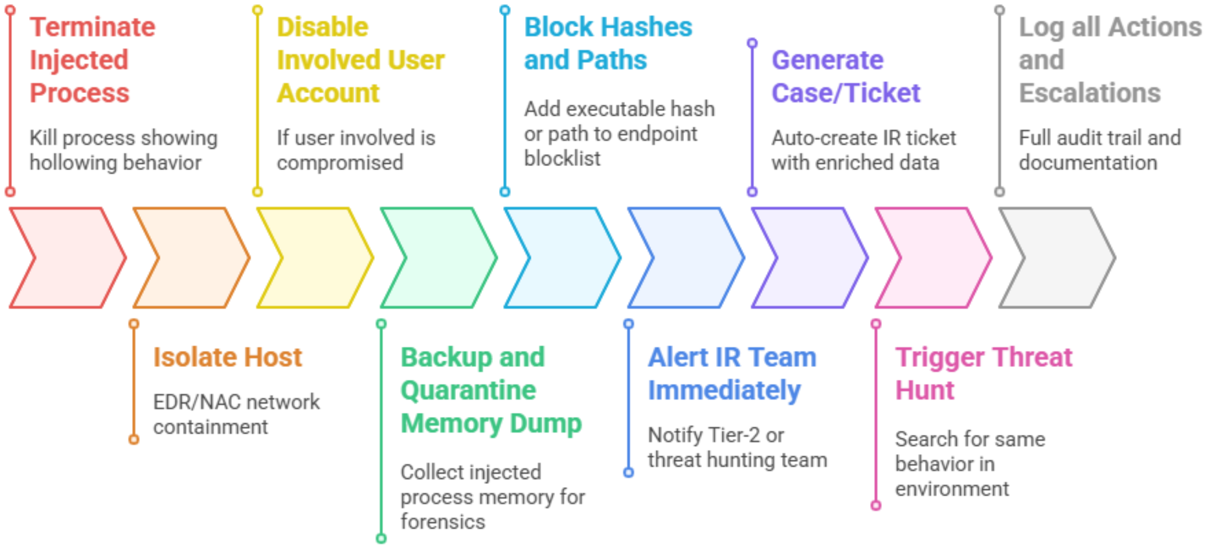
5. Automated Enrichment

Enrichment Task	Description
User & Host Attribution	Who triggered the behavior and from which machine
Injected & Target Process Analysis	Target PID, name, command line, hash, parent PID
Command-Line Inspection	Was --suspended or --hidden used?
API Call Chain Correlation	Map CreateProcess, VirtualAllocEx, WriteProcessMemory, ResumeThread sequence
Check Image Consistency	Compare memory-loaded image vs. file on disk
Hash & Binary Reputation	Check binary in VirusTotal, internal intel

6. Automated Response Play

Step	Action
1. Terminate Injected Process	Kill process showing hollowing behavior
2. Isolate Host	EDR/NAC network containment
3. Disable Involved User Account	If user involved is compromised
4. Backup and Quarantine Memory Dump	Collect injected process memory for forensics
5. Block Hashes and Paths	Add executable hash or path to endpoint blocklist
6. Alert IR Team Immediately	Notify Tier-2 or threat hunting team
7. Generate Case/Ticket	Auto-create IR ticket with enriched data
8. Trigger Threat Hunt	Search for same behavior in environment
9. Log all Actions and Escalations	Full audit trail and documentation

Incident Response Protocol for Process Hollowing



Terminate Injected Process
Kill process showing hollowing behavior

Isolate Host
EDR/NAC network containment

Disable Involved User Account
If user involved is compromised

Backup and Quarantine Memory Dump
Collect injected process memory for forensics

Block Hashes and Paths
Add executable hash or path to endpoint blocklist

Alert IR Team Immediately
Notify Tier-2 or threat hunting team

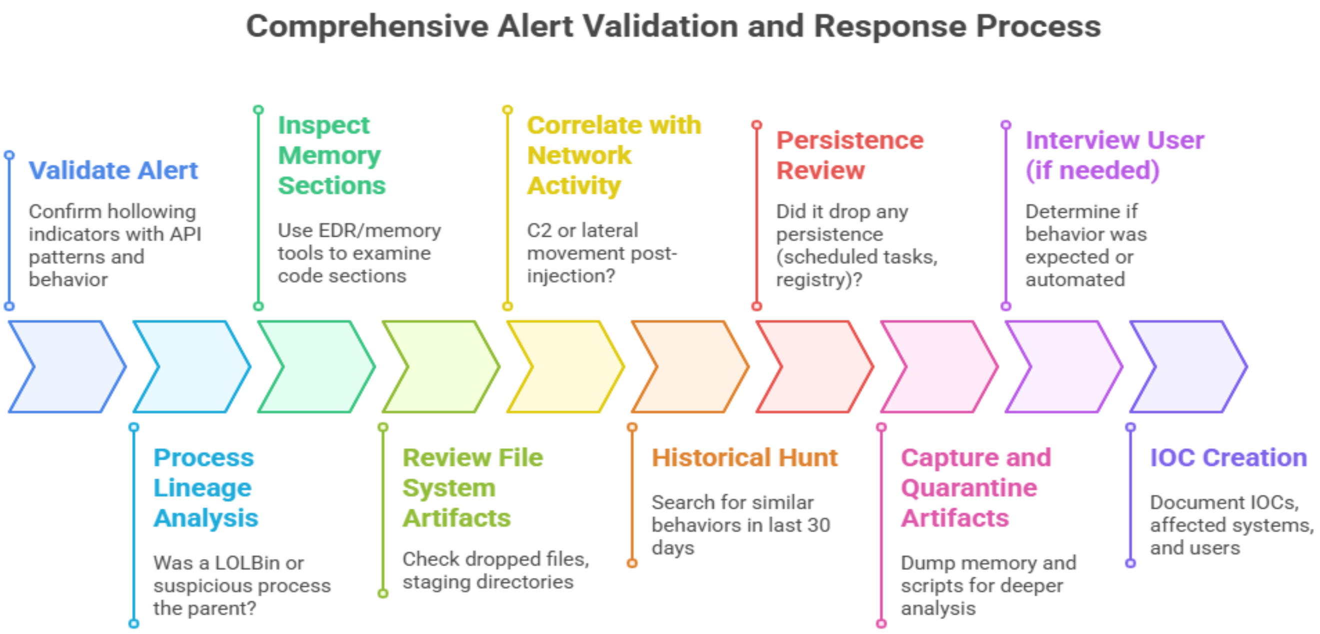
Generate Case/Ticket
Auto-create IR ticket with enriched data

Trigger Threat Hunt
Search for same behavior in environment

Log all Actions and Escalations
Full audit trail and documentation

7. Investigation Checklist

Step	Description
1. Validate Alert	Confirm hollowing indicators with API patterns and behavior
2. Process Lineage Analysis	Was a LOLBin or suspicious process the parent?
3. Inspect Memory Sections	Use EDR/memory tools to examine code sections
4. Review File System Artifacts	Check dropped files, staging directories
5. Correlate with Network Activity	C2 or lateral movement post-injection?
6. Historical Hunt	Search for similar behaviors in last 30 days
7. Persistence Review	Did it drop any persistence (scheduled tasks, registry)?
8. Capture and Quarantine Artifacts	Dump memory and scripts for deeper analysis
9. Interview User (if needed)	Determine if behavior was expected or automated
10. IOC Creation	Document IOCs, affected systems, and users



Comprehensive Alert Validation and Response Process

Validate Alert
Confirm hollowing indicators with API patterns and behavior

Process Lineage Analysis
Was a LOLBin or suspicious process the parent?

Inspect Memory Sections
Use EDR/memory tools to examine code sections

Review File System Artifacts
Check dropped files, staging directories

Correlate with Network Activity
C2 or lateral movement post-injection?

Historical Hunt
Search for similar behaviors in last 30 days

Persistence Review
Did it drop any persistence (scheduled tasks, registry)?

Capture and Quarantine Artifacts
Dump memory and scripts for deeper analysis

Interview User (if needed)
Determine if behavior was expected or automated

IOC Creation
Document IOCs, affected systems, and users

8. Playbook Notes

- **Sysmon Configured for ImageLoad and Remote Thread Creation** (Event IDs 7, 8, 10).
- Understand T1055.012 attack sequences — it's a frequent APT tactic.
- Block use of suspicious LOLBins (regsvr32, mshta, etc.) from temp folders.
- Use memory analysis tools (Volatility, Rekall) to inspect injected processes.
- Monitor for behaviors like **high memory entropy**, **API call anomalies**, and **unsigned memory regions**.
- Baseline normal process execution trees — e.g., svchost.exe should not launch other apps.