

CHAPTER 14

Proactive Activities

Chapter Overview

01

Threat Hunting

Assume breach — hypothesis-driven search for undetected adversaries

02

Adversary Emulation

Atomic Red Team, CALDERA, and purple team exercises

Threat Hunting

Assume breach — go find them

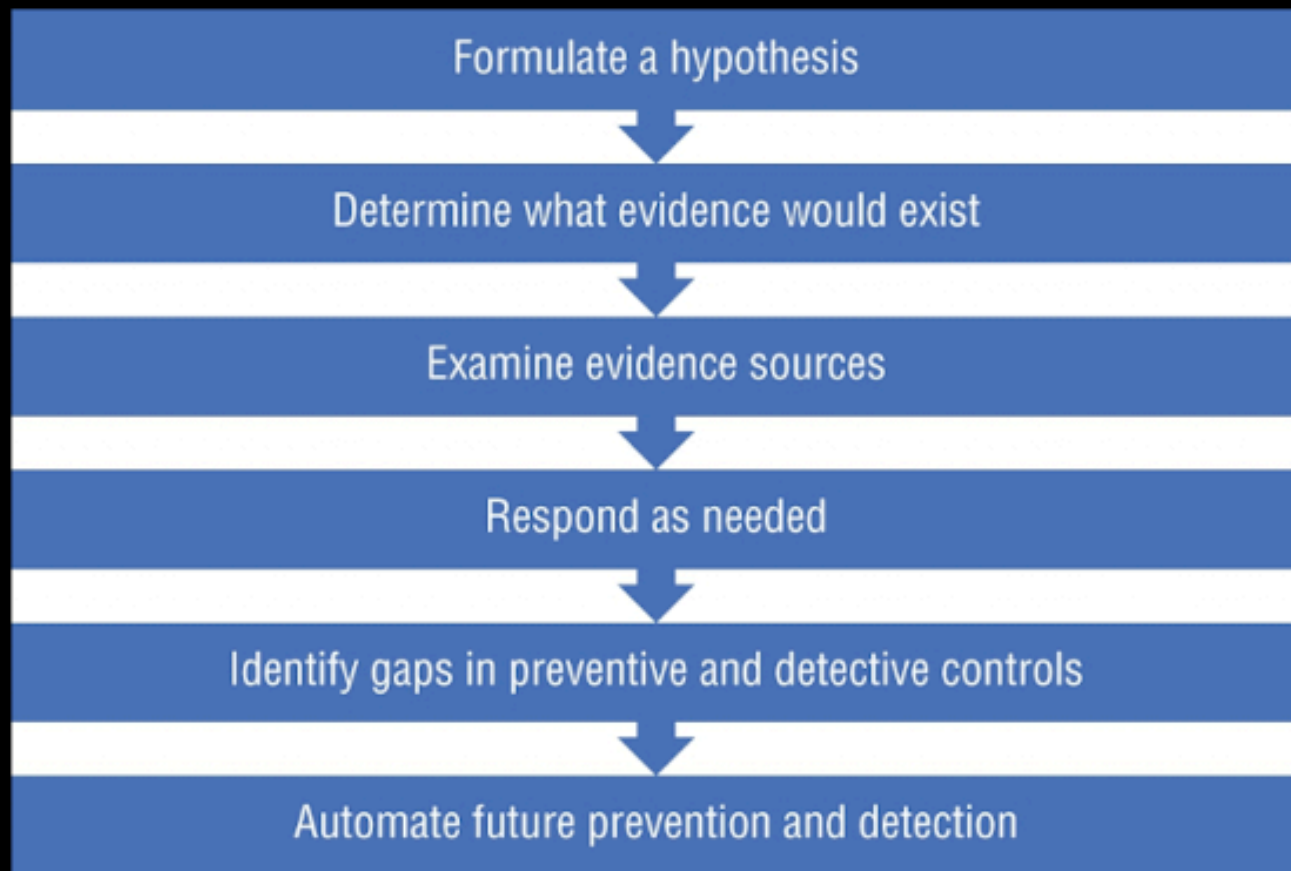


Figure 14.1: The threat-hunting process

Threat Hunting Fundamentals

Threat hunting: proactively search for adversaries that evaded automated detection

Core assumption: you are already compromised — the question is where and for how long

Alert-driven response is reactive; hunting is proactive — different mindset entirely

MITRE ATT&CK: every technique is a potential hunt hypothesis (1400+ techniques)

Hunt loop: Form hypothesis → Identify data sources → Execute hunt → Analyze → Document → Automate

Output: either 'found something' (escalate to IR) or 'tuned detection' (new detection rule)

Maturity: hunts that find nothing still produce value — they become detection content

Building Hunt Hypotheses

Hypothesis format: 'Attackers using [technique] would leave [indicator] in [data source]'

Example: 'Attackers using Kerberoasting would generate 4769 events with RC4 encryption'

Example: 'C2 beaconing via HTTP would appear as regular-interval connections in proxy logs'

Threat intelligence: recent TTPs from threat reports → immediate hunt hypotheses

Crown jewel analysis: what data is most valuable? Work backwards from target to access path

Gap analysis: which ATT&CK techniques have NO detection coverage? Hunt those first

Prior incidents: the attacker may still be present — use the IR timeline as a hunt guide

Process & Memory Hunt Examples

Process-Based Hunts

Parent-child anomalies: winword.exe → powershell.exe (T1566, T1059)

LOLBins with networking: certutil.exe -urlcache, regsvr32.exe, mshta.exe

Long-lived PowerShell: processes running >8 hours without user interaction

Hollow processes: process image path doesn't match loaded module on disk

Unusual LSASS access: non-system processes with PROCESS_VM_READ on lsass

Memory & Injection Hunts

Executable memory outside image-backed regions (malfind in Volatility)

DLLs loaded from unusual paths: AppData, Temp, ProgramData

Threads with start address pointing to heap or anonymous memory

Process with network connection but no corresponding listening socket

svchost.exe with no -k argument in command line

Network & Credential Hunt Examples

Network-Based Hunts

Beaconing: automated C2 check-ins at fixed intervals (1–60 min jitter)

Long connections: hours-long sessions on port 443 with low byte transfer

DNS tunneling: high-entropy subdomains, unusually large TXT record responses

Newly registered domains: C2 domains registered within last 30 days

Internal east-west: workstation → workstation on SMB — never normal

Credential Abuse Hunts

4769 RC4 spike: bulk TGS requests from single host in short window

4648 Explicit credential use: lateral movement tools trigger this constantly

Admin account logons on workstations other than their assigned host

Service accounts logging on interactively (Type 2) — should never happen

NTLM Type 3 responses from hosts that should be using Kerberos only

Data Sources & Hunt Infrastructure

EDR telemetry: process, file, registry, network events at endpoint level

SIEM / log aggregation: centralized query across all hosts simultaneously

DNS logs: full query/response history — gold mine for C2 and data exfil hunting

Proxy / web gateway logs: outbound HTTP/HTTPS URL and byte-count details

NetFlow / IPFIX: connection metadata without full payload — covers the whole network

Zeek logs: rich protocol-specific fields from network taps

Velociraptor VQL: run any hunt query across all endpoints simultaneously in real time

Key principle: if the data source doesn't exist, the hunt can't run — build visibility first

Measuring Hunt Effectiveness

MTTD (Mean Time to Detect): how long attackers dwell before discovery — aim to reduce it

Detection coverage: percentage of ATT&CK techniques with at least one active detection

Hunt yield: ratio of investigations opened per hunt executed — track over time

New detections: every hunt that finds something should produce at least one new detection rule

False positive rate: high FPR burns analyst time — tune aggressively

Hunt cadence: high-priority techniques should be hunted weekly; others monthly/quarterly

Document everything: hypothesis, data source, query, results, and disposition

Hunt Platform Setup & Tooling

Minimum viable hunt stack: SIEM + EDR + DNS logs + NetFlow

Elastic Stack: free, scalable, Kibana for visualization — most common open-source stack

Splunk: market-leading SIEM — powerful query language (SPL), high cost at scale

Microsoft Sentinel: cloud-native SIEM — deep Azure/M365 integration, pay per GB

Velociraptor: complements SIEM with live endpoint query capability

Threat intel integration: MISP or commercial TI feed populates indicators for hunt queries

SOAR platforms: orchestrate repetitive hunt steps (XSOAR, Splunk SOAR)

Start simple: 5 good hunt queries executed weekly beats 100 automations never reviewed

Reconnaissance	Resource Development	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
11 techniques	8 techniques	11 techniques	17 techniques	23 techniques	14 techniques	47 techniques	17 techniques	34 techniques	9 techniques	17 techniques	18 techniques	9 techniques	15 techniques
Active Scanning (3)	Acquire Access	Content Injection	Cloud Administration Command	Account Manipulation (7)	Abuse Elevation Control Mechanism (6)	Abuse Elevation Control Mechanism (6)	Adversary-in-the-Middle (4)	Account Discovery (4)	Exploitation of Remote Services	Adversary-in-the-Middle (4)	Application Layer Protocol (3)	Automated Exfiltration (7)	Account Access Removal
Gather Victim Host Information (4)	Acquire Infrastructure (8)	Drive-by Compromise	Command and Scripting Interpreter (13)	BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)	Brute Force (4)	Application Window Discovery	Internal Spearphishing	Archive Collected Data (2)	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction (2)
Gather Victim Identity Information (2)	Compromise Accounts (5)	Exploit Public-Facing Application	Container Administration Command	Boot or Logon Autostart Execution (14)	Account Manipulation (7)	Account Manipulation (7)	Credentials from Password Stores (4)	Browser Information Discovery	Lateral Tool Transfer	Remote Service Session Hijacking (2)	Content Injection	Exfiltration Over Alternative Protocol (2)	Data Encrypted for Impact
Gather Victim Network Information (5)	Compromise Infrastructure (8)	External Remote Services	Deploy Container	Boot or Logon Initialization Scripts (5)	Boot or Logon Autostart Execution (14)	Boot or Logon Autostart Execution (14)	Exploitation for Credential Access	Cloud Infrastructure Discovery	Remote Services (2)	Automated Collection	Data Encoding (2)	Defacement (2)	Data Manipulation (2)
Gather Victim Org Information (4)	Develop Capabilities (4)	Hardware Additions	ESXi Administration Command	Cloud Application Integration	Boot or Logon Initialization Scripts (5)	Boot or Logon Initialization Scripts (5)	Forced Authentication	Cloud Service Dashboard	Replication Through Removable Media	Browser Session Hijacking	Data Obfuscation (5)	Exfiltration Over C2 Channel	Disk Wipe (2)
Phishing for Information (4)	Establish Accounts (2)	Phishing (4)	Replication Through Removable Media	Compromise Host Software Binary	Create or Modify System Process (2)	Create or Modify System Process (2)	Deobfuscator/Decode Files or Information	Cloud Service Discovery	Software Deployment Tools	Clipboard Data	Dynamic Resolution (2)	Exfiltration Over Other Network Medium (1)	Email Bombing
Search Closed Sources (2)	Obtain Capabilities (7)	Stage Capabilities (5)	Input Injection	Create Account (3)	Domain or Tenant Policy Modification (2)	Domain or Tenant Policy Modification (2)	Deploy Container	Container and Resource Discovery	Taint Shared Content	Data from Cloud Storage	Encrypted Channel (2)	Exfiltration Over Physical Medium (1)	Endpoint Denial of Service (4)
Search Open Technical Databases (3)	Supply Chain Compromise (2)	Trusted Relationship	Inter-Process Communication (2)	Create Account (3)	Event Triggered Execution (14)	Event Triggered Execution (14)	Direct Volume Access	Debugger Evasion	Use Alternate Authentication Material (4)	Data from Information Repositories (5)	Fallback Channels	Exfiltration Over Web Service (4)	Financial Theft
Search Open Websites/Domains (2)	Valid Accounts (4)	Native API	Poisoned Pipeline Execution	Event Triggered Execution (14)	Escape to Host	Escape to Host	Domain Trust Discovery	Device Driver Discovery	Multi-Factor Authentication (Process) (8)	Data from Configuration Repository (2)	Hide Infrastructure	Exfiltration Over Web Service (4)	Firmware Corruption
Search Threat Vendor Data	Wi-Fi Networks	Scheduled Task/Job (3)	Scheduled Task/Job (3)	Exclusive Control	Exploitation for Privilege Escalation	Exploitation for Privilege Escalation	Email Spoofing	Domain Trust Discovery	Multi-Factor Authentication Request Generation	Data from Information Repositories (5)	Ingress Tool Transfer	Scheduled Transfer	Inhibit System Recovery
Search Victim-Owned Websites	Serverless Execution	Shared Modules	Serverless Execution	External Remote Services	Hijack Execution Flow (12)	Hijack Execution Flow (12)	Execution Guardrails (2)	File and Directory Permissions Modification (2)	Network Sniffing	Data from Local System	Multi-Stage Channels	Transfer Data to Cloud Account	Network Denial of Service (2)
	Software Deployment Tools	System Services (2)	Software Deployment Tools	Implant Internal Image	Process Injection (12)	Process Injection (12)	Exploitation for Defense Evasion	Hide Artifacts (14)	OS Credential Dumping (3)	Data from Network Shared Drive	Non-Application Layer Protocol	Resource Hijacking (4)	Resource Hijacking (4)
	Modify Authentication Process (2)	User Execution (3)	System Services (2)	Modify Authentication Process (2)	Scheduled Task/Job (3)	Scheduled Task/Job (3)	Exploitation for Defense Evasion	Hijack Execution Flow (12)	Network Service Discovery	Data from Removable Media	Non-Standard Port	Service Stop	Service Stop
	Modify Registry	Windows Management Instrumentation	User Execution (3)	Modify Registry	Valid Accounts (4)	Valid Accounts (4)	Impair Defenses (2)	Impair Defenses (2)	Network Share Discovery	Data Staged (2)	Protocol Tunneling	System Shutdown/Reboot	System Shutdown/Reboot
	Office Application Startup (5)	Power Settings	Windows Management Instrumentation	Office Application Startup (5)	Indicator Removal (10)	Indicator Removal (10)	Indirect Command Execution	Impersonation	Network Sniffing	Email Collection (3)	Proxy (4)		
								Impersonation	Password Policy Discovery	Input Capture (4)	Remote Access Tools (2)		
								Steal or Forge Kerberos Tickets (3)	Peripheral Device Discovery	Screen Capture	Traffic Signaling (2)		
									Permission Groups Discovery (2)	Video Capture	Web Service (3)		

Adversary Emulation

Test your defenses before attackers do

Adversary Emulation vs Red Team vs Pentest

Penetration test: find vulnerabilities — scope is systems, goal is access

Red team: simulate full adversary campaign — scope is detection & response capability

Adversary emulation: execute specific TTPs to test specific detections — scoped, collaborative

Purple team: red + blue working together with immediate feedback — learning exercise

Key advantage of emulation: blue team learns what detection gaps exist and fixes them in real time

Prerequisites: logging must be in place before emulation — otherwise nothing to measure

ATT&CK Navigator: visualize coverage gaps before planning an emulation exercise

Atomic Red Team

Framework Overview

Open-source library mapping to every ATT&CK technique (1400+ atomic tests)

Each test: Prerequisites, Executor, Cleanup — self-contained and reversible

PowerShell framework: Install-AtomicRedTeam;
Import-Module Invoke-AtomicRedTeam

Invoke-AtomicTest T1059.001: runs all PowerShell execution tests

Invoke-AtomicTest T1059.001 -TestNumbers 1: runs a single specific test

Using Atomics in Purple Team

Step 1: choose ATT&CK technique with weak/no detection coverage

Step 2: run atomic test → observe what telemetry is generated

Step 3: write SIEM/EDR detection rule targeting the observed artifacts

Step 4: run atomic again → verify the detection fires

Step 5: add test to regression suite — re-run after every detection change

MITRE CALDERA

Platform & Architecture

Automated adversary emulation platform from MITRE — open source

Server: Python-based web application with REST API and UI

Agents (Sandcat): deployed to target systems, beacon to CALDERA server

Abilities: individual TTPs mapped to ATT&CK — hundreds included

Adversary profiles: ordered list of abilities = simulated threat actor

Running an Exercise

Step 1: deploy Sandcat agent on test endpoint (staged, not production)

Step 2: select adversary profile (e.g., APT3, ransomware, credential theft)

Step 3: start operation — CALDERA executes each ability, records results

Step 4: review operation report — which abilities succeeded vs. blocked

Step 5: for each undetected success, create detection and re-run

Closing the Loop — From Hunt to Detection

Every successful hunt finding becomes a permanent detection rule in the SIEM or EDR

Every successful emulation finding that wasn't detected becomes an alert to build

Document the ATT&CK technique, data source, query, and false positive rate for each rule

Detection-as-code: store detection logic in version control alongside the code it monitors

Regression testing: run atomic tests after every detection platform change to verify nothing broke

Threat intel integration: ingest new TTP reports → update hunt backlog → emulate within 30 days

The loop: Threat Intel → Hunt Hypothesis → Emulate → Detect → Hunt again

Reporting Hunt Findings

Audience: two reports — executive (1 page) and technical (as detailed as needed)

Executive report: threats found, dwell time, business risk, recommended actions

Technical report: hypothesis, data sources, query logic, findings, IOCs, investigation steps

Negative hunts: 'found nothing' is a valid result — document what was looked at

ATT&CK mapping: include technique IDs for all findings — enables coverage tracking over time

IOC dissemination: share file hashes, IPs, and domains to SIEM/EDR blocking within 24 hours

Trend reporting: monthly summary — hunts run, yield rate, new detections created

Value demonstration: translate technical findings into business risk for leadership buy-in

Integrating Hunting with Incident Response

Detection gap: every IR alert starts as a hunt hypothesis — close the loop

During IR: while one team contains, another hunts for additional compromised systems

Hunt feeds IR: hypotheses proven true during hunting are handed to IR for full investigation

IR feeds hunts: TTPs discovered during IR become hunt hypotheses for the next quarter

Post-incident hunt: after remediation, re-hunt using attacker's known TTPs — verify full eviction

Shared tooling: hunt and IR teams share the same SIEM, EDR, and log infrastructure

Joint exercises: purple team exercises train both hunt and IR capabilities simultaneously

Continuous loop: Incident → Document TTPs → Hunt for TTPs → Detect → Improve

Conclusion

Reactive-only security guarantees you discover breaches long after damage is done

Threat hunting systematically reduces attacker dwell time — every day of hunting saves months of cleanup

MITRE ATT&CK provides a universal language and priority framework for both hunts and emulations

Atomic Red Team turns ATT&CK into executable tests — your detection coverage is only as good as what you've tested

CALDERA automates multi-stage adversary campaigns — stress tests detection and response together

Complete the loop: intelligence → hypothesis → hunt → emulate → detect → repeat

Knowledge Check

The image shows the Kahoot! logo in white, bold, sans-serif font, centered on a purple background. The background is a blurred image of a school hallway with a grid ceiling and several doors.

Kahoot!