Lecture Title: Threat Hunting – Proactive Cyber Defense

---

1. Introduction to Threat Hunting

1.1 Definition

Threat hunting is a proactive, hypothesis-driven process of searching through networks, endpoints, and datasets to detect advanced threats that evade traditional security controls.

Unlike reactive security (alerts), hunting assumes: "**The adversary is already inside**."

1.2 Key Objectives

- Detect Advanced Persistent Threats (APT)
- Identify unknown/zero-day attacks
- Reduce dwell time
- Improve detection engineering

1.3 Threat Hunting vs Other Security Functions

Function	Nature	Trigger	Tools
SOC Monitoring	Reactive	Alerts	SIEM
Incident Response	Reactive	Incident	EDR, Forensics
Threat Intelligence	Predictive	External intel	TIP
Threat Hunting	Proactive	Hypothesis	SIEM + EDR + Logs

---

2. Threat Hunting Frameworks

2.1 MITRE ATT&CK Framework

MITRE ATT&CK

- Maps attacker behavior into:
  - Tactics (e.g., Persistence, Lateral Movement)
  - Techniques (e.g., Pass-the-Hash)

Example:

- Tactic: Credential Access
- Technique: LSASS Dumping

2.2 Cyber Kill Chain

Lockheed Martin Cyber Kill Chain

Stages:

1. Reconnaissance

2. Weaponization
3. Delivery
4. Exploitation
5. Installation
6. Command & Control
7. Actions on Objectives

2.3 Pyramid of Pain

Pyramid of Pain

Level	Difficulty for Attacker
Hash Values	Easy
IP Addresses	Easy
Domain Names	Medium
Artifacts	Hard
Tools	Harder
TTPs	**Very Hard**

Focus hunting on TTPs, not just IOCs.

3. Threat Hunting Methodology

3.1 Hypothesis-Driven Hunting

Example hypothesis: **"An attacker may be using PowerShell for lateral movement."**

3.2 Hunting Lifecycle

Step 1: Hypothesis Creation

- Based on:
  - Threat Intelligence
  - Past incidents
  - ATT&CK techniques

Step 2: Data Collection

Sources:

- Endpoint logs (EDR)
- Network logs (Firewall, IDS)
- Authentication logs (AD)
- Application logs

Step 3: Data Analysis

Tools:

- Splunk
- ELK Stack
- Microsoft Defender for Endpoint

Techniques:

- Statistical anomaly detection
- Behavioral analysis
- Correlation queries

Step 4: Detection & Investigation

- Validate suspicious activity
- Pivot across logs

Step 5: Response & Mitigation

- Isolate host
- Block IOC
- Patch vulnerability

Step 6: Documentation & Feedback

- Improve SIEM rules
- Update playbooks

4. Types of Threat Hunting

4.1 Structured Hunting

- Based on ATT&CK

4.2 Unstructured Hunting

- Triggered by anomaly or alert

4.3 Situational Hunting

- Based on environment context (e.g., banking system)

5. Real-World Banking Use Cases

Use Case 1: Suspicious PowerShell Execution

Scenario:

- Attacker executes encoded PowerShell

Hunting Query (Splunk):

```
index=windows EventCode=4688
| search powershell.exe
| where like(CommandLine,"%EncodedCommand%")
```

Use Case 2: Lateral Movement Detection
Indicators:

- Multiple login attempts across servers
- Use of admin credentials

Query:

```
index=security EventCode=4624 Logon_Type=3
| stats count by Account_Name, ComputerName
| where count > 10
```

Use Case 3: Data Exfiltration
Indicators:

- Large outbound traffic

Query:

```
index=network sourcetype=firewall
| stats sum(bytes_out) by src_ip
| where sum(bytes_out) > 100000000
```

Use Case 4: Credential Dumping
Indicators:

- Access to LSASS (Local Security Authority Subsystem Service) process

6. Hands-On Lab
Lab Title: Detect Suspicious PowerShell Activity

6.1 Lab Environment
Tools:

- Splunk or ELK
- Windows VM
- Sysmon logs

6.2 Setup
Step 1: Install Sysmon

```
sysmon -i sysmonconfig.xml
```

Step 2: Generate Malicious Activity
Run:

```
powershell.exe -EncodedCommand SQBtACAAZQB2AGkAbAA=
```

6.3 Log Analysis
Query:

```
index=sysmon EventCode=1
| search powershell.exe
| table _time, Computer, CommandLine
```

6.4 Investigation Steps

1. Identify encoded command
2. Decode Base64
3. Check parent process
4. Correlate with user login

6.5 Expected Outcome

Students should:

- Detect suspicious PowerShell
- Understand attacker behavior
- Build detection rule

7. Advanced Hunting Techniques

7.1 Behavioral Analytics

- Detect anomalies vs baseline

7.2 Machine Learning-Based Hunting

- UEBA systems

7.3 Threat Intelligence Integration

- IOC enrichment
- Feed correlation

8. Key Challenges

- Data overload
- False positives
- Skill gap
- Tool limitations

9. Best Practices

- Automate repetitive tasks
- Focus on high-value assets
- Use ATT&CK mapping
- Continuous improvement

10. Conclusion

Threat hunting is:

- Proactive

- Intelligence-driven
- Critical for modern SOC maturity

Core Principle
Focus hunting on TTPs (Tactics, Techniques, Procedures), not just IOCs (Indicators of Compromise)

1. Definitions (Precision Required)

1.1 Indicators of Compromise (IOCs)

- Observable artifacts of an attack
- Examples:
    - Malicious IP (e.g., 185.XX.XX.10)
    - File hash (MD5/SHA256)
    - Domain name

These are forensic traces, not behavior.

1.2 TTPs (Adversary Behavior)

Referenced in MITRE ATT&CK

- Tactics → Why (objective)
- Techniques → How (method)
- Procedures → Implementation (specific tools/steps)

Example:

- Tactic: Credential Access
- Technique: OS Credential Dumping
- Procedure: Using Mimikatz

2. Why IOCs Are Weak (Operational Reality)

Problem 1: Easy to Change

- IP → rotated in minutes
- Domain → regenerated
- Hash → changes with minor modification

Attackers adapt faster than signature updates.

Problem 2: Reactive Nature

- IOCs are discovered after compromise
- They represent known attacks only

Problem 3: Low Context

- Seeing an IP ≠ understanding attacker intent
- No visibility into attack lifecycle

3. Why TTPs Are Strong (Strategic Advantage)

3.1 Hard to Change

- Attackers must still:
  - Move laterally
  - Escalate privileges
  - Execute commands

Behavior is constrained by system architecture.

3.2 Detect Unknown Attacks

- Even zero-day malware:
  - Still spawns processes
  - Still accesses memory
  - Still communicates

3.3 Aligns with Adversary Thinking

- Enables behavioral detection
- Supports proactive hunting

4. Practical Comparison

Aspect	IOC-Based Hunting	TTP-Based Hunting
Nature	Reactive	Proactive
Longevity	Short-lived	Long-lasting
Detection Type	Signature	Behavioral
Coverage	Known threats	Known + Unknown
Evasion Resistance	Low	High

5. Real Example (Banking Environment)

Scenario: Malware Infection

IOC-Based Approach

- Detect:
  - File hash = XYZ123
  - IP = 10.10.10.5

Problem: Attacker changes hash → detection fails

TTP-Based Approach

Hunt for:

- Suspicious PowerShell execution
- Credential dumping behavior
- Unusual lateral movement

Example behaviors:
index=windows EventCode=4688
| search powershell.exe
| where like(CommandLine,"%EncodedCommand%")
Even if malware changes, behavior remains similar.

6. Mapping Example Using ATT&CK

Tactic	Technique	Hunt Focus
Execution	PowerShell	Encoded commands
Credential Access	LSASS Dumping	Memory access
Lateral Movement	SMB/WinRM	Remote logins

7. Advanced Insight
Detection Engineering Evolution
Level 1: IOC Matching
- Antivirus signatures
- Blacklists

Level 2: Rule-Based Detection
- SIEM correlation

Level 3: TTP-Based Hunting
- ATT&CK mapping
- Behavioral analytics

Level 4: Adaptive Detection
- UEBA / ML

8. Threat Hunter Mindset Shift
Instead of asking: "**Is this IP malicious?"**
Ask: "**Is this behavior normal for this system?**"

9. Practical Hunting Hypotheses (TTP-Focused)
1. "Attackers may use PowerShell for execution"
2. "Adversaries may attempt credential dumping from LSASS"
3. "Lateral movement may occur via abnormal SMB sessions"

10. Key Takeaway
- IOCs tell you WHAT happened
- TTPs tell you HOW and WHY it happened

And in threat hunting: Understanding HOW is far more powerful than knowing WHAT

- IOC = “Fingerprint of a criminal”
- TTP = “Criminal’s method of operation (MO)”
- Criminal can wear gloves (change IOC) But method (TTP) often remains consistent

Lecture Title: TTPs (Tactics, Techniques, and Procedures) in Cybersecurity

---

1. Introduction to TTPs

1.1 Definition

TTPs represent the behavioral patterns of adversaries, describing:

- Why an attack is conducted (Tactics)
- How it is executed (Techniques)
- How exactly it is implemented (Procedures)

1.2 Origin & Standardization

TTPs are formally structured in:

- MITRE ATT&CK

This framework is the global standard for:

- Threat intelligence
- Detection engineering
- Threat hunting

2. Breakdown of TTP Components

2.1 Tactics (The WHY)

Tactics represent the attacker's objective at a particular stage.
Examples:

- Initial Access
- Execution
- Persistence
- Privilege Escalation
- Defense Evasion
- Credential Access
- Lateral Movement
- Exfiltration

Think of tactics as: "What is the attacker trying to achieve right now?"

2.2 Techniques (The HOW)

Techniques describe how attackers achieve a tactic.
Example:

- Tactic: Credential Access
- Technique: OS Credential Dumping

2.3 Procedures (The IMPLEMENTATION)

Procedures are:

- Specific tools or commands used

Example:

- Technique: Credential Dumping
- Procedure:
  - Using Mimikatz
  - Using ProcDump

3. ATT&CK Matrix (Core Learning Model)

3.1 Structure

- Rows → Techniques
- Columns → Tactics

3.2 Example Mapping

Tactic	Technique	Procedure
Execution	PowerShell	Encoded command
Persistence	Registry Run Keys	Modify registry
Credential Access	LSASS Dumping	Mimikatz

4. TTPs Across Attack Lifecycle

4.1 Mapping with Kill Chain

Lockheed Martin Cyber Kill Chain

Kill Chain Stage	TTP Example
Delivery	Phishing
Exploitation	Script execution
Installation	Persistence
C2	Beaconing
Action	Data exfiltration

5. Why TTPs Are Critical (Advanced Insight)

5.1 Stability of Behavior

- IPs change
- Malware changes
- BUT behavior persists

TTPs are hard to evade

5.2 Enables Detection of Unknown Threats

- Zero-day malware still:

- Executes processes
- Accesses memory
- Communicates

5.3 Foundation of Modern SOC

TTPs drive:

- Threat hunting
- Detection engineering
- Incident response

6. TTP vs IOC vs IOA (Critical Comparison)

Concept	Focus	Level
IOC	Artifact	Low
IOA	Behavior pattern	Medium
TTP	Strategy	High

Relationship:

- TTP → defines attacker behavior
- IOA → detects behavior
- IOC → confirms compromise

7. Real-World TTP Examples (Banking Context)

Example 1: Phishing-Based Attack

TTP Chain:

- Initial Access → Phishing
- Execution → Macro-enabled document
- Persistence → Registry modification

Example 2: Credential Theft

TTP Chain:

- Privilege Escalation
- Credential Access → LSASS dump
- Lateral Movement → SMB

Example 3: Data Exfiltration

TTP Chain:

- Collection → Sensitive files
- Exfiltration → HTTPS channel

8. Detection Engineering Using TTPs

8.1 Example: PowerShell Abuse

Technique: Command execution via PowerShell
Detection Query:

```
index=windows EventCode=4688
| search powershell.exe
| where like(CommandLine,"%EncodedCommand%")
```

8.2 Example: Lateral Movement

```
index=security EventCode=4624 Logon_Type=3
| stats count by Account_Name, ComputerName
| where count > 10
```

8.3 Example: Data Exfiltration

```
index=network
| stats sum(bytes_out) by src_ip
| where sum(bytes_out) > threshold
```

9. Threat Hunting Using TTPs

9.1 Hypothesis Example "**Attackers may use PowerShell for lateral movement.**"

9.2 Hunting Approach

- Identify logs
- Build queries
- Correlate events
- Validate anomalies

9.3 Key Principle

Hunt for:

- Behavior
- Sequence
- Anomalies

NOT just indicators.

10. Advanced Concepts

10.1 TTP Profiling (Adversary Attribution)

- Nation-state actors
- Cybercriminal groups

Each has: Unique TTP patterns

10.2 Detection Maturity Model

1. IOC-based detection
2. Rule-based detection
3. IOA-based detection
4. TTP-based detection
5. AI-driven detection

10.3 Mapping to Controls

TTP	Control
Credential Dumping	EDR monitoring
Lateral Movement	Network segmentation
Exfiltration	DLP

11. Challenges with TTP-Based Detection

- Requires skilled analysts
- Needs large data visibility
- Complex correlation logic
- High implementation cost

12. Best Practices

- Use ATT&CK mapping
- Continuously update detection rules
- Integrate threat intelligence
- Automate detection (SOAR)

13. Practical Lab (Mini Exercise)

Lab Title: Detect TTP – PowerShell Execution
Objective: Detect suspicious execution behavior

Steps:

1. Generate activity:

```
powershell.exe -EncodedCommand SQBtACAAZQB2AGkAbAA=
```

2. Query logs:

```
index=windows EventCode=4688
| search powershell.exe
```

3. Analyze:

- Command line
- Parent process
- User context

Outcome:

- Identify TTP
- Understand attacker behavior

14. Key Takeaways

- TTPs represent attacker behavior
- More reliable than IOC-based detection

Essential for:

- Threat hunting
- SOC operations
- Incident response

Use this analogy:

- IOC = "Fingerprint"
- IOA = "Suspicious movement"
- TTP = "Criminal strategy"

"Cyber defense maturity is achieved when organizations detect behavior, not just artifacts."

Indicator of Attack (IOA)

1. What is IOA (Indicator of Attack)?

1.1 Definition

Indicators of Attack (IOA) represent behavioral evidence that an attack is in progress, focusing on how an adversary operates, rather than static artifacts.
IOA answers: **"What suspicious activity pattern indicates an ongoing attack?"**

1.2 Key Characteristics

- Behavioral (not static)
- Context-aware
- Time-sequenced (chain of events)
- Often mapped to MITRE ATT&CK

1.3 Simple Example

IOC:

- File hash = abc123

IOA:

- A process:
  - Spawns powershell.exe
  - Executes encoded command
  - Connects to external IP

This sequence is the IOA.

2. IOA vs IOC – Precise Comparison

Feature	IOC (Indicator of Compromise)	IOA (Indicator of Attack)
Nature	Static artifact	Behavioral pattern
Focus	What was used	How attack is executed
Timing	After compromise	During attack
Detection Type	Signature-based	Behavior-based
Evasion Resistance	Low	High
Context Awareness	Minimal	High
Zero-day Detection	Weak	Strong

3. Why IOA is Superior (Deep Technical Justification)

3.1 Resilience Against Evasion

IOC Limitation:

- Hash → easily modified
- IP → rotated

- Domain → regenerated

IOA Advantage:

- Attackers must execute behavior
    - Process creation
    - Privilege escalation
    - Lateral movement

These are hard to avoid.

3.2 Detects Unknown & Zero-Day Attacks

Even if malware is new:

- It still:
    - Executes code
    - Calls system APIs
    - Communicates externally

IOA detects behavior, not signature.

3.3 Captures Attack Lifecycle

IOA aligns with:

- Lockheed Martin Cyber Kill Chain
- MITRE ATT&CK

Example:

- Initial access → Execution → Persistence → Exfiltration

IOC only captures fragments, IOA captures flow.

3.4 Enables Proactive Threat Hunting

IOC: "Check if bad IP exists"
IOA: "Hunt for suspicious login + process + network behavior"

IOA supports hypothesis-driven hunting

3.5 Reduces False Positives (When Properly Modeled)

- IOC: IP may be shared → false alert
- IOA: multi-condition correlation → higher fidelity

Example:

- Single login → normal
- Multiple failed logins + privilege escalation → suspicious

4. Real-World IOA Examples (Banking Context)

4.1 Credential Dumping (High-Risk)

IOA Pattern:

- Process accesses LSASS memory
- Followed by unusual authentication activity

Detection Logic:

```
index=sysmon EventCode=10
| search TargetImage="lsass.exe"
```

4.2 Lateral Movement

IOA Pattern:

- Multiple logins across servers
- Same user, short time window

```
index=security EventCode=4624 Logon_Type=3
| stats count by Account_Name, ComputerName
| where count > 10
```

4.3 Living-off-the-Land Attack (LOLBins)

IOA Pattern:

- powershell.exe OR cmd.exe
- Encoded or obfuscated commands

```
index=windows EventCode=4688
| search powershell.exe
| where like(CommandLine,"%EncodedCommand%")
```

4.4 Data Exfiltration

IOA Pattern:

- Large outbound traffic
- Unusual destination

```
index=network
| stats sum(bytes_out) by src_ip
| where sum(bytes_out) > threshold
```

5. IOA Construction Model

5.1 Event-Based Model

IOA =
Event 1 + Event 2 + Event 3 + Context

Example:

- Process creation
- Network connection
- Privilege escalation

5.2 Behavioral Chain
Initial Access → Execution → Persistence → Exfiltration

Each stage contributes to IOA.

6. IOA in Modern Security Tools
EDR Platforms
- Microsoft Defender for Endpoint
- CrowdStrike Falcon

Use IOA heavily:
- Behavioral detection rules
- Attack chain visualization

SIEM + UEBA
- Correlates:
  - User behavior
  - Network anomalies
  - Endpoint activity

7. IOC vs IOA vs TTP (Critical Distinction)

Concept	Focus	Level
IOC	Artifact	Low-level
IOA	Behavior Pattern	Mid-level
TTP	Adversary Strategy	High-level

Relationship:
- TTP → defines behavior
- IOA → detects behavior
- IOC → confirms artifact

8. Advanced Insight (Detection Engineering Perspective)
Detection Maturity Model
1. IOC-based detection
2. Rule-based correlation
3. IOA-based detection
4. TTP-based hunting
5. AI/Behavioral analytics

Modern SOC maturity requires IOA + TTP alignment

9. Practical Threat Hunting Hypothesis (IOA-Based)
Instead of: "Check malicious IP"
Use: "If a user logs in from unusual location AND executes PowerShell AND connects externally → possible compromise"

10. Key Takeaways

- IOA is behavioral and contextual
- Detects ongoing attacks, not just past compromise
- Resistant to evasion techniques

Critical for:

- Threat hunting
- EDR detection
- Advanced SOC

Explain hierarchy like this:

- IOC → "Evidence left behind"
- IOA → "Suspicious behavior in action"
- TTP → "Attacker strategy"

"Modern cybersecurity is no longer about identifying bad files, but understanding bad behavior."