**Guide to Threat Hunting: IT vs. ICS Environments**

**What is Threat Hunting?**

**Threat hunting** is a proactive, hypothesis-driven process where defenders look for signs of adversary behavior **that may have evaded traditional detection tools**.

It differs from automated alerting or incident response because:

* You’re actively **searching for threats** rather than waiting for alerts
* It’s led by **human analysis and intuition**
* It often uncovers **novel or low-and-slow attacks**

**The Threat Hunting Process: Key Phases**

1. **Hypothesis Development**
2. **Data Collection & Scope Definition**
3. **Data Exploration & Querying**
4. **Pivoting & Correlation**
5. **IOC/Behavioral Pattern Identification**
6. **Validation & Documentation**
7. **Feedback Loop (Detection Engineering)**

**Step 1: Crafting a Hunting Hypothesis**

A good hypothesis answers the question:

“If an attacker were doing **X**, what would that look like in our data?”

**🔹 Good Hypothesis Examples:**

* *“An attacker may have used Modbus function code 5 (Write Single Coil) during off-hours.”*
* *“If a compromised engineering workstation is beaconing to a C2 server, we should see unusual HTTP/DNS requests from that host.”*
* *“If APT33 compromised a machine, we may see unauthorized RDP followed by abnormal Modbus writes.”*

**🔹 Tips for Hypothesis Crafting:**

* **Base it on threat intel** (e.g., MITRE ATT&CK for ICS, APT reports)
* Include **who/what/where/when/why**
* Make it **testable** using log data you have access to

**IT vs. ICS Threat Hunting – Key Differences**

| **Aspect** | **Traditional IT Environment** | **ICS/OT Environment** |
| --- | --- | --- |
| **Priority Assets** | Servers, endpoints, email, web services | PLCs, RTUs, IEDs, HMIs, SCADA servers |
| **Protocols** | HTTP, SMB, RDP, SSH, DNS | Modbus, DNP3, OPC UA, Profinet, CIP |
| **Detection Tools** | EDR, NIDS, SIEM | Passive monitoring, protocol parsers (e.g., Zeek) |
| **Risks of Response** | Can isolate/reboot systems | Must avoid disrupting critical operations |
| **Log Sources** | Endpoint logs, firewall, AD, NetFlow | ICS protocol logs, serial converters, sensors |
| **Timing** | IT attacks can happen 24/7 | ICS attacks often occur off-hours to evade notice |
| **Baselines** | Baseline can vary daily | ICS traffic is highly deterministic and regular |

**Why ICS Threat Hunting Is Different**

ICS environments:

* Have **limited tolerance for scanning or probing** (don’t run active tools!)
* Include **proprietary or legacy protocols** with limited security features
* Rely heavily on **physical process control**, meaning attacks may cause real-world harm
* Have **strict availability requirements** — false positives or disruptions can halt production

As a result, ICS threat hunters must:

* Use **passive, log-based methods**
* Rely on **protocol-specific behavioral anomalies**
* Focus on **“how” devices communicate**, not just “who” or “when”

**Crafting an ICS-Specific Hypothesis**

Example:

"If an attacker has compromised an internal engineering workstation, they may attempt to issue unauthorized Modbus commands to PLCs during off-hours."

To investigate:

* Filter for log\_type: modbus.log
* Look for source IPs that are **not the known master**
* Focus on function codes: 5, 6, 15, 16
* Filter timestamps: 01:00 – 06:00 UTC

**Tools for Threat Hunting in ELK**

In your ELK (Elastic) stack, use:

| **Tool** | **Use Case** |
| --- | --- |
| **Discover** | Log browsing and field-based filtering |
| **Dashboards** | Visualizing patterns and anomalies |
| **Data Views** | Segmenting logs by type (e.g., modbus) |
| **Search KQL/DSL** | Building powerful, filterable queries |

**Practical Example: ICS APT33 Threat Hunt**

**Hypothesis**:

"If APT33 is operating from a compromised internal container (192.168.3.30), we will see unauthorized Modbus and DNP3 activity targeting ICS outstations after business hours."

**Hunting Strategy**:

1. Filter log\_type: "modbus.log" or "dnp3.log"
2. Filter id.orig\_h: 192.168.3.30
3. Focus on function codes: 5, 6, 15, 16
4. Set time filter to off-hours (e.g., midnight to 6am)
5. Pivot into conn.log to see related sessions or scanning
6. Check dns.log or http.log for signs of beaconing to external C2 IPs

**Correlate Findings**:

* If a write command (e.g., Modbus func 5) is sent from a non-master IP…
* And RDP or DNS traffic was seen from that IP shortly before…
* You have an IOC chain tied to lateral movement and exfiltration.

**Takeaways**

* **Hypothesis-driven hunting** gives direction to your search
* **ICS hunting requires caution and context** — your target is often safety-critical infrastructure
* Use the **predictability of ICS traffic** to your advantage
* Focus on **unauthorized actions**, **abnormal timing**, and **non-standard sources**