Research Report

FORESCOUT

From Events to TTPs:

Maturing OT Incident Response with MITRE ATT&CK® for ICS

by Forescout Research Labs

Table of Contents

Executive Summary

The **growing threat landscape for operational technology (OT)** networks, exemplified by a number of recent ransomware attacks [1], has prompted critical infrastructure **organizations to better prepare themselves for impactful cyber incidents**. To do this, stakeholders responsible for critical infrastructure and services are maturing their security operations centers (SOCs) and increasing their use of cyber threat intelligence (CTI). Many now consider **adversarial Tactics, Techniques and Procedures (TTPs) to be their most valuable CTI tool** [2].

The recently released **MITRE ATT&CK for Industrial Control Systems (ICS)** framework compiles OT-specific TTPs collected from real-world data and provides a common nomenclature for industrial security practitioners to better prepare for, detect and respond to cyber incidents.

In this paper, we show how an OT network monitoring and intrusion detection solution (IDS), combined with the ATT&CK for ICS framework, can **enhance an organization's OT incident response process** in three phases of the Incident Response lifecycle from NIST [12]:

- **Preparation:** Why a network-based IDS is a crucial data source that must be put in place to detect events of interest, how these events can be mapped to TTPs and how to evaluate detection against a set of standard TTPs. To do this, we map OT-specific event types generated by Forescout eyeInspect™ to ATT&CK for ICS, and then demonstrate how to correlate them with actual detected events using traffic from a Capture the Flag (CTF) competition.

- **Detection & Analysis:** We demonstrate how mapping of events to TTPs helps bridge the semantic gap between attackers acting strategically to achieve their goals and defenders processing low-level events to detect attacks. In this section, we present a reproduction of a real incident and how an analyst can proceed step-by-step during an investigation.

- **Containment, Eradication & Recovery:** We show how events and TTPs can be forwarded to Security Orchestration, Automation and Response (SOAR) tools to achieve an orchestrated response.

We'll wrap up by discussing how Forescout enables a **holistic, OT-specific cybersecurity strategy from detection to response** by leveraging cutting-edge detection technology, associating it with community-driven knowledge in the form of TTPs and integrating with existing tools to reduce mean time to respond (MTTR).

1. Why Use MITRE ATT&CK for ICS?

One of the main challenges in cybersecurity is the **semantic gap between attackers and defenders** [3]. While **attackers think strategically** and employ different **TTPs** to achieve their goals, **defenders must process low-level events** that are generated by **IDS** that only provide information about small steps within larger attacks [4]. A version of this problem has been summarized by Microsoft Distinguished Engineer John Lambert: *“Defenders think in lists. Attackers think in graphs. As long as this is true, attackers win.”* [5].

Tactics refer to the **objectives** that attackers want to achieve, such as gaining initial access into a network. **Techniques** are the **actions** that attackers take to achieve a tactical objective, such as exploit public-facing applications. **Procedures** are specific implementation examples of Techniques used by adversaries, such as using sqlmap for SQL injection.

To close this gap, stakeholders adopt models **that allow them to better understand, contextualize and stop cyberattacks, and invest in tools that operationalize these models into the Incident Response lifecycle**.

1.1. Understanding Attacker Models

In the last decade, there have been **several attempts** at modeling attack lifecycles and attacker behavior, such as the **Lockheed Martin Cyber Kill Chain** [6], the **Mandiant Attack Lifecycle** [7], the **SANS ICS Cyber Kill Chain** [8] and the MITRE ATT&CK[1] Framework [9]. The **last** is arguably the **most comprehensive attempt thus far**, in terms of coverage of attacker behavior, and **most successful**, in terms of industry adoption.

[1] ATT&CK stands for Adversarial Tactics, Techniques, and Common Knowledge.

According to MITRE, ATT&CK is a “**knowledge base of tactics and techniques based on real-world observations of adversaries**”. In the ATT&CK framework, the tactics and techniques are presented in different matrices, each modeling attacker behavior in a **specific domain**. Until the end of 2019, there were three available matrices:

- PRE-ATT&CK, focusing on pre-compromise activities.
- Enterprise, focusing on activities to compromise Windows, macOS, Linux and cloud systems.
- Mobile, focusing on activities to compromise Android and iOS devices.

These matrices are mostly applicable to IT devices and networks, where the final goals of attackers are usually data exfiltration or financial gain. Thus, **OT and ICS** network defenders, who deal with threats targeting the availability, integrity and safety of their industrial processes, **were left without a proper attacker model**.

The **ATT&CK for ICS** matrix was officially released in **January 2020** [10]. ATT&CK for ICS **extends the previous framework** with three important **tactics**, namely Inhibit Response Function, Impair Process Control and Impact, which model the kind of destructive goal that ICS attackers are known for, while dropping three enterprise-focused tactics, namely Privilege Escalation, Credential Access and Exfiltration. Several **new OT-focused techniques** were also identified, including Data Historian Compromise, Engineering Workstation Compromise, and Modify Control Logic.

Now that OT defenders have a community-accepted attacker model and list of TTPs, that will be continuously maintained and updated, it’s time to integrate this intelligence into the tools used in their **Incident Response** processes.

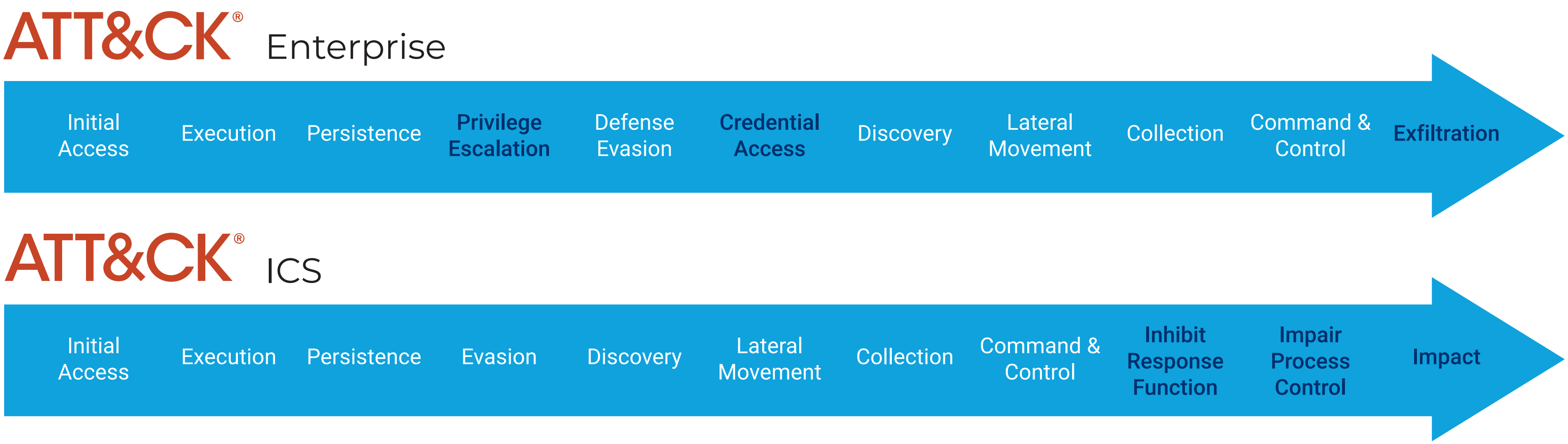


Figure 1: Tactics in ATT&CK for Enterprise vs. ATT&CK for ICS.

1.2. The Incident Response Lifecycle

The NIST Computer Security Incident Handling Guide [11] divides the Incident Response Lifecycle into **four phases**:

1. The **Preparation** phase involves the establishment of an incident response capability and the prevention of incidents by ensuring sufficient security.
2. The **Detection & Analysis** phase involves the timely detection of relevant events via IDS and other tools, as well as their escalation into incidents after initial analysis.
3. The **Containment, Eradication & Recovery** phase involves the steps taken to respond to these incidents.
4. Finally, the **Post-Incident Activity** phase involves the lessons learned from an incident and how to improve the existing process.

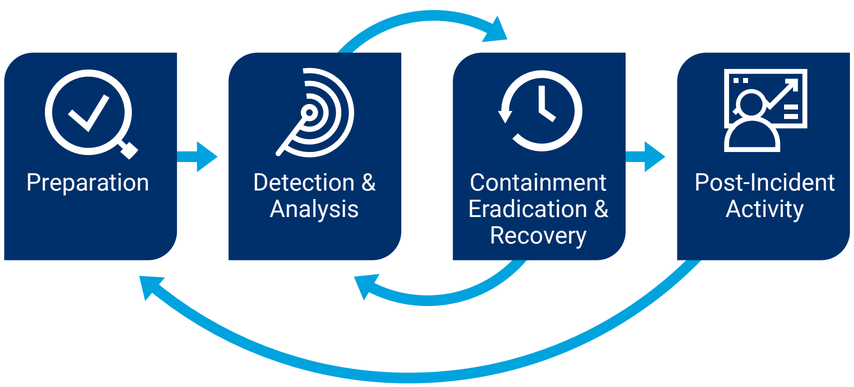


Figure 2: The Incident Response Lifecycle [11].

Although this lifecycle was not designed around OT networks, it's flexible enough to also be useful for them. However, there are certain challenges that can arise when using this lifecycle for OT networks, including:

1. **The need for specialized detection**. Being prepared for an incident means at the very least being able to detect it. OT events may be specific both to a type of device and to vendor-specific devices.
2. **Increasing analyst fatigue**. This fine-grained detection could lead to thousands of events generated in a single day, making it difficult to determine which ones are indicative of real incidents. The number of events also increases with the growth of monitored sites. Imagine how many events would be detected in all the substations of an electric utility company or in all distribution centers of a major online retailer. Finding correlated events is dependent on contextual information coming from disparate tools.
3. **Limited OT expertise**. Most incident responders lack sufficient OT knowledge to be able to efficiently and effectively respond to an incident. This includes knowing what components could be the next targets of an ongoing attack and how an adversary might hide in the network. OT-specific tools often fail to provide this knowledge for analysts.

1.3. Maturing Incident Response Strategy for OT

Using the NIST Incident Response Lifecycle as a guide, we will demonstrate how eyeInspect and ATT&CK for ICS can help to address these challenges and **enhance an organization's OT incident response process**:

1. **Preparation:** Better prepare for OT incidents by using a network-based IDS that has extensive detection coverage for TTPs in ATT&CK for ICS.
2. **Detection & Analysis:** Analyzing detected OT incidents using ATT&CK for ICS provides guidance for security teams on where to focus attention and how to proceed with an investigation to better understand the context of single events.
3. **Containment, Eradication and Recovery:** Using a common nomenclature helps defenders better understand what an attacker has already achieved, what their next moves might be and the potential impact of an incident. We also show how forwarding events to a SOAR tool enables playbook-oriented response [12].

4. **These improvements are not only qualitative** in terms of what attacks can be detected and how well-prepared an organization can be, but also quantitative in terms of a reduction in the MTTR to incidents, thus saving analyst hours.

2. Turning Events into TTPs

Preparation for cyber incidents involves **setting up a detection and response capability** to help **prevent incidents in the first place**. We will **not discuss incident prevention** in this paper, but it comprises effective application of security controls, including maintaining an up-to-date asset inventory, proper network segmentation and frequent risk assessment.

Though incident prevention is extremely important, the security community has recognized that incidents will happen and can happen to any organization that is sufficiently targeted. Thus, it's becoming more common to adopt an "assume breach" mindset [13], where **the most important thing is to make sure that even when an incident takes place, it can be detected and stopped as soon as possible**.

Therefore, our focus in the Preparation phase is **setting up the necessary tools that allow us to detect, investigate and stop these incidents**. We accomplish this by:

1. **Detecting Events:** We show why a network-based intrusion detection system (NIDS) is a crucial data source that must be put in place to detect malicious events.

2. **Mapping Events to TTPs:** We discuss how to map the events from a NIDS to the TTPs they represent to gain more context into what stage of an attack a detected event could represent.

3. **Evaluating Detection:** We analyze detection capabilities in a security assessment scenario, to make sure that the tactics we expect are detected in a realistic situation.

2.1. Detecting Events

To detect a potentially malicious event, a dedicated tool must first **observe the event from a data source**, be it network metadata (via NetFlow), full network traffic (via port mirroring and Deep Packet Inspection), or process- and filesystem-related events on a host (via sysmon for Windows hosts). The tool must then provide an automatic alert about what it has detected.

Forescout researchers analyzed the information available on MITRE's website about data sources used to detect techniques and assets affected by techniques. The results are shown in Figure 3 and Figure 4.

Our analysis shows that the new ATT&CK for ICS matrix lists 41 data sources, with **network-based sources detecting the vast majority of techniques**, as shown in **Figure 3**. There are 81 unique techniques in total, described in detail in Section 2.2, 67 of which list their data sources.

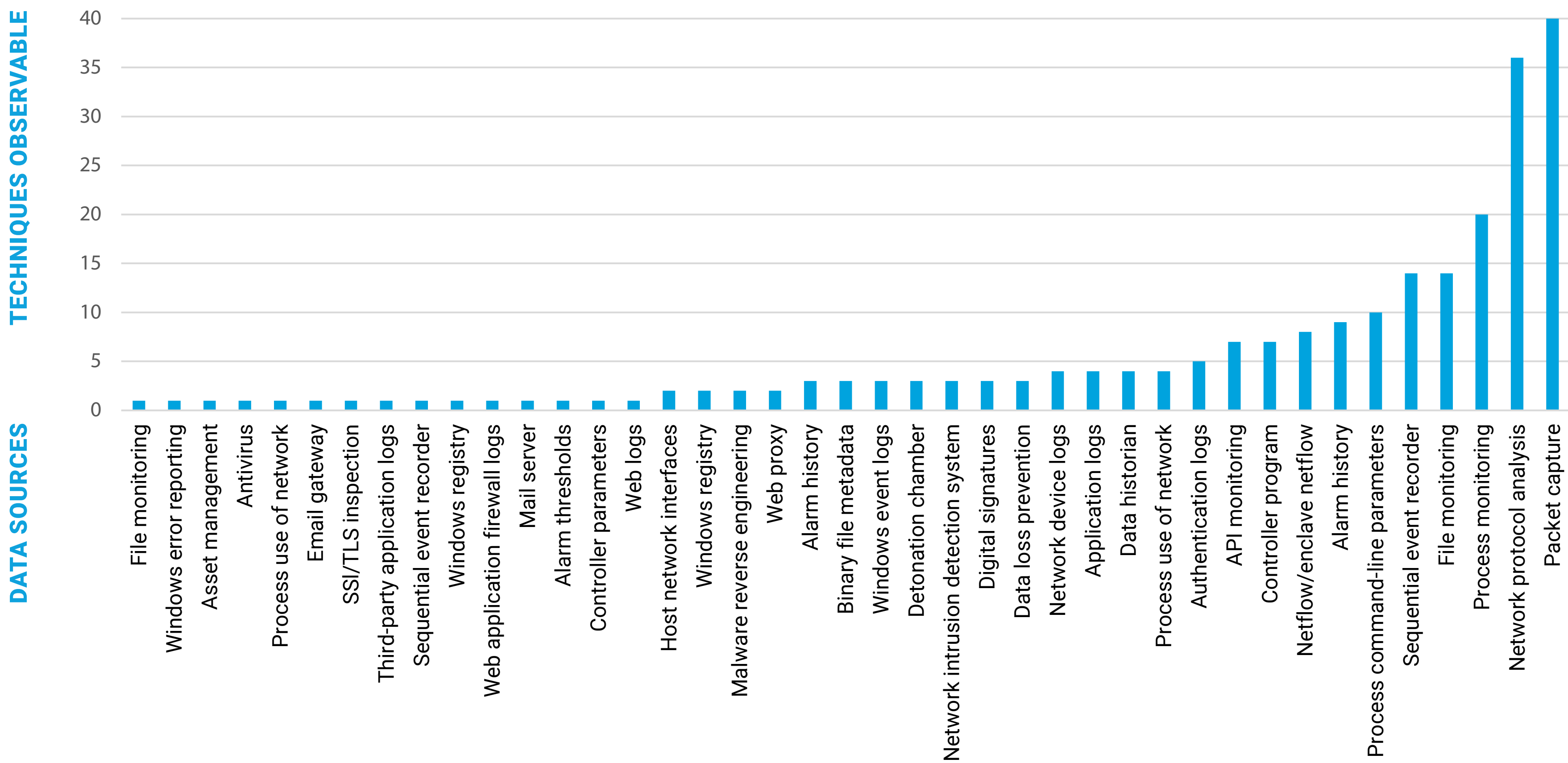


Figure 3: Number of techniques observable from each data source.

Another important characteristic of OT networks is that **host-based events are difficult to gather, which translates to being both more expensive and easier to miss for critical infrastructure and services providers**. This is because critical embedded devices – such as Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs) and Intelligent Electronic Devices (IEDs) – have **no support for security tools**. Even Windows-based computers, such as engineering workstations, have constraints on the applications they can run. Moreover **critical embedded devices are still affected by the majority of the techniques** in ATT&CK for ICS, as shown in **Figure 4**.

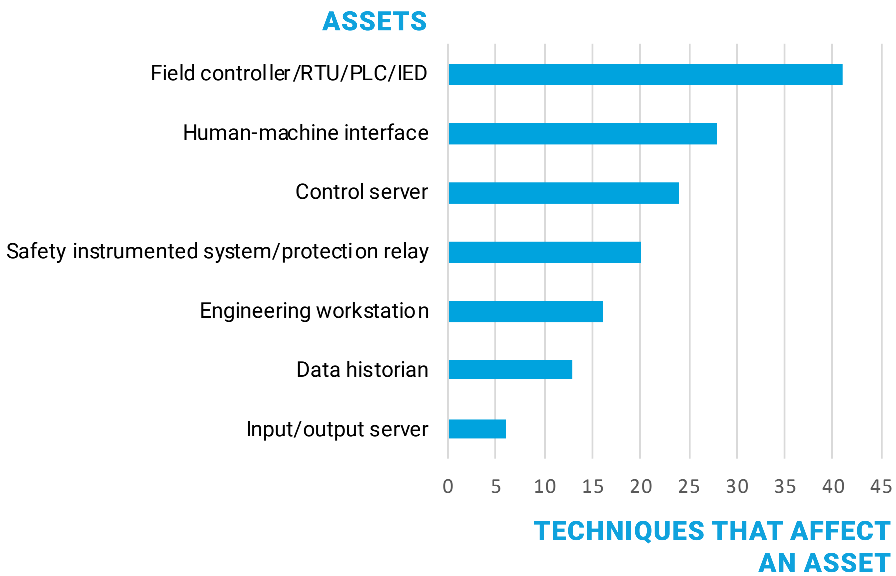


Figure 4: Number of techniques affecting each type of asset.

EyeInspect has **five dedicated threat detection engines**, each capable of analyzing network traffic in a specialized way to detect both cybersecurity and operational events that could be indicative of an attempted attack. The detection engines are the following:

- **Basic Engines (Malformed Packets, Port Scan, Man in the Middle)**. These basic engines detect very specific networking issues and attacks, including the presence of malformed packets in hundreds of OT and IT protocols, like the malformed SMB packets used in WannaCry. They also detect the occurrence of port scanning activity, which is used by attackers to do reconnaissance of network-enabled devices, as well as attempts to establish man-in-the-middle attacks.

- **Industrial Threat Library (ITL)**. This engine contains extensive checks for OT-specific threat indicators that work out of the box and are based on Forescout's expertise from more than a decade in OT cybersecurity. Examples of detected threats include potentially dangerous operations executed via industrial protocols, misconfigured or misbehaving devices, use of insecure protocols and possible data breaches.

- **Local Area Network Communication Patterns (LANCP)**. This engine learns and monitors communication patterns in the local network, such as which devices communicate with which other devices and over which protocols. After a learning period, the engine can raise events when it detects a new communication pattern in the network.

- **Deep Packet Behavioral Inspection (DPBI)**. This engine learns and monitors the contents of detailed packet fields for specialized OT protocols communicated between two devices. After a period of learning, the engine can raise events when it detects a packet with anomalous content being transferred between devices.

- **Scripting Engine**. This engine allows the user (and Forescout analysts) to quickly extend eyeInspect's detection capabilities whenever new threats emerge (see our response to URGENT/11) or to cater to customized threat detection scenarios.

Besides those five dedicated threat detection engines, **eyeInspect also supports the detection of known Indicators of Compromise** (IoCs) in the form of YARA rules and malicious file hashes, as well as blacklisted client applications, IP addresses and domains.

In total, eyeInspect has **more than 1,200 event types** produced by these detection engines, which enables fine-grained detection of multiple kinds of threats manifesting in **more than 130 IT and OT-specific networking protocols**.

2.2. Mapping Events to TTPs

Mapping the events raised by these IDS tools to the TTPs in ATT&CK for ICS lets security teams gain more context into what an event in the network means in terms of an attack and active defense.

Figure 6 shows a **high-level view of all ATT&CK for ICS tactics and techniques**. The Figure uses the traditional matrix visualization of the ATT&CK framework. **Each column is a tactic** and **each cell** under a column **is a technique** that can be used to accomplish the goals of that tactic. The ATT&CK for ICS matrix contains **81 unique techniques** grouped into **11 tactics**. (Note that some techniques appear in more than one tactic.)

Initial Access	Execution	Persistence	Evasion	Discovery	Lateral Movement	Collection	Command and Control	Inhibit Response Function	Impair Process Control	Impact
Data Historian Compromise	Change Program State	Hooking	Exploitation for Evasion	Control Device Identification	Default Credentials	Automated Collection	Commonly Used Port	Activate Firmware Update Mode	Brute Force I/O	Damage to Property
Drive-by Compromise	Command-Line Interface	Module Firmware	Indicator Removal on Host	I/O Module Discovery	Exploitation of Remote Services	Data from Information Repositories	Connection Proxy	Alarm Suppression	Change Program State	Denial of Control
Engineering Workstation Compromise	Execution through API	Program Download	Masquerading	Network Connection Enumeration	External Remote Services	Detect Operating Mode	Standard Application Layer Protocol	Block Command Message	Masquerading	Denial of View
Exploit Public-Facing Application	Graphical User Interface	Project File Infection	Rogue Master Device	Network Service Scanning	Program Organization Units	Detect Program State		Block Reporting Message	Modify Control Logic	Loss of Availability
External Remote Services	Man in the Middle	System Firmware	Rootkit	Network Sniffing	Remote File Copy	I/O Image		Block Serial COM	Modify Parameter	Loss of Control
Internet Accessible Device	Program Organization Units	Valid Accounts	Spoof Reporting Message	Remote System Discovery	Valid Accounts	Location Identification		Data Destruction	Module Firmware	Loss of Productivity and Revenue
Replication Through Removable Media	Project File Infection		Utilize/Change Operating Mode	Serial Connection Enumeration		Monitor Process State		Denial of Service	Program Download	Loss of Safety
Spearphishing Attachment	Scripting					Point & Tag Identification		Device Restart/ Shutdown	Rogue Master Device	Loss of View
Supply Chain Compromise	User Execution					Program Upload		Manipulate I/O Image	Service Stop	Manipulation of Control
Wireless Compromise						Role Identification		Modify Alarm Settings	Spoof Reporting Message	Manipulation of View
						Screen Capture		Modify Control Logic	Unauthorized Command Message	Theft of Operational Information
								Program Download		
								Rootkit		
								System Firmware		
								Utilize/Change Operating Mode		

Figure 5: MITRE ATT&CK for ICS tactics and techniques.

There are three impact techniques explicitly mentioned by MITRE as not being detectable, since they are related to non-technical goals of adversaries. These are "Damage to Property," "Loss of Productivity and Revenue" and "Theft of Operational Information." Some other techniques are not directly detectable via network monitoring, but some of their associated cause and effects (such as file transfers) may be observed by eyeInspect. These are "Masquerading," "Rootkit," "Screen Capture," and "Wireless Compromise." The other techniques can be detected by eyeInspect's detection engines and contextual information.

As an example, we mapped **1,270 unique built-in event types from eyeInspect 4.1 to ATT&CK techniques** that do not require specific contextual information, so that every time one of these events is observed in the network it can be directly mapped to a technique. The various techniques covered by eyeInspect are mapped below.

Initial Access	Execution	Persistence	Evasion	Discovery	Lateral Movement	Collection	Command and Control	Inhibit Response Function	Impair Process Control	Impact
Data Historian Compromise	Change Program State	Hooking	Exploitation for Evasion	Control Device Identification	Default Credentials	Automated Collection	Commonly Used Port	Activate Firmware Update Mode	Brute Force I/O	Damage to Property
Drive-by Compromise	Command-Line Interface	Module Firmware	Indicator Removal on Host	I/O Module Discovery	Exploitation of Remote Services	Data from Information Repositories	Connection Proxy	Alarm Suppression	Change Program State	Denial of Control
Engineering Workstation Compromise	Execution through API	Program Download	Masquerading	Network Connection Enumeration	External Remote Services	Detect Operating Mode	Standard Application Layer Protocol	Block Command Message	Masquerading	Denial of View
Exploit Public-Facing Application	Graphical User Interface	Project File Infection	Rogue Master Device	Network Service Scanning	Program Organization Units	Detect Program State		Block Reporting Message	Modify Control Logic	Loss of Availability
External Remote Services	Man in the Middle	System Firmware	Rootkit	Network Sniffing	Remote File Copy	I/O Image		Block Serial COM	Modify Parameter	Loss of Control
Internet Accessible Device	Program Organization Units	Valid Accounts	Spoof Reporting Message	Remote System Discovery	Valid Accounts	Location Identification		Data Destruction	Module Firmware	Loss of Productivity and Revenue
Replication Through Removable Media	Project File Infection		Utilize/Change Operating Mode	Serial Connection Enumeration		Monitor Process State		Denial of Service	Program Download	Loss of Safety
Spearphishing Attachment	Scripting					Point & Tag Identification		Device Restart/ Shutdown	Rogue Master Device	Loss of View
Supply Chain Compromise	User Execution					Program Upload		Manipulate I/O Image	Service Stop	Manipulation of Control
Wireless Compromise						Role Identification		Modify Alarm Settings	Spoof Reporting Message	Manipulation of View
						Screen Capture		Modify Control Logic	Unauthorized Command Message	Theft of Operational Information
								Program Download		
								Rootkit		
								System Firmware		
								Utilize/Change Operating Mode		

Figure 6: Techniques covered by eyeInspect.

- Blue – Automatically detected through eyeInspect built in alerts from existing database of event types
- Grey – Detected through eyeInspect's proprietary event detection engines and other contextual information
- Orange – Outside of the scope of network detection tools

Since eyeInspect has multiple detection engines, including dedicated anomaly- and signature-based detection for IT and OT protocols (as described in Section 2.1), we can map several event types to the same technique. This multi-factor detection is important because it provides a level of redundancy so that if a new evasion capability allows attackers to bypass one type of detection for a certain technique, other types of detection can pick it up.

Figure 7 shows the **number of eyeInspect event types that can be used to detect each tactic**. Notice that the tactics towards the end of the attack lifecycle are the ones with the most detection events. These are also the tactics with the highest potential disruption impact (except for "Lateral Movement").

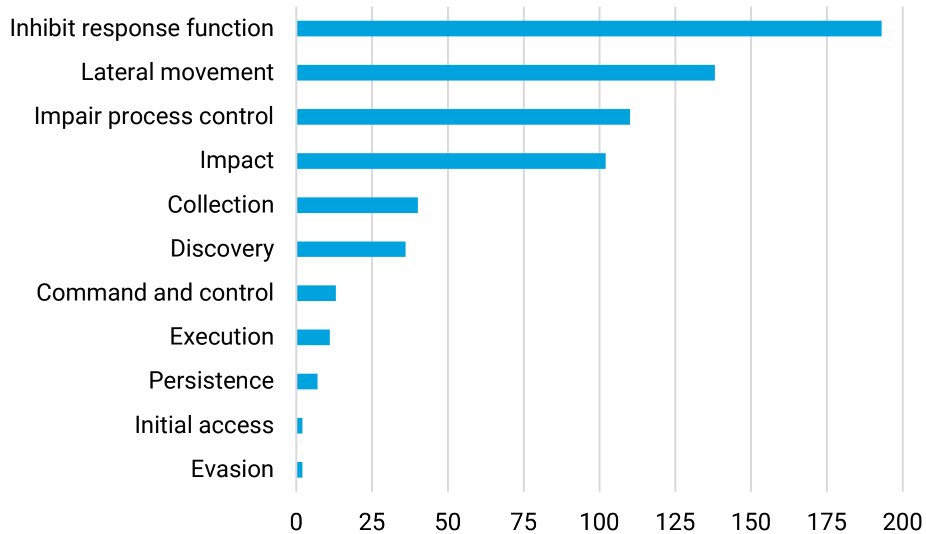


Figure 7: Number of eyeInspect event types that can detect techniques within each tactic.

Table 1 details the mapping between eyeInspect event types and ATT&CK techniques. Each row presents a technique (second column), the number of eyeInspect event types that can detect it (third column) and the detection engines that produce those event types (fourth column). Recall that the techniques not listed in Table 1 are also detectable by eyeInspect, there are just no specific event types mapped to them.

Table 1: Examples of techniques with Specific Detections

Tactic	Technique	eyeInspect Event Types	eyeInspect Detection Engines
Initial Access	Exploit Public-Facing Application	2	ITL
	Replication Through Removable Media	1	ITL
Execution	Command-Line Interface	3	ITL
	Man in the Middle	7	MITM
Persistence	Program Download	23	ITL, Script
	System Firmware	14	ITL
	Valid Accounts	35	ITL
Evasion	Indicator Removal on Host	1	ITL
	Rogue Master Device	4	LANCP, ITL
	Utilize/Change Operating Mode	2	ITL
Discovery	Control Device Identification	9	ITL, Script
	I/O Module Discovery	1	ITL
	Network Service Scanning	17	ITL, LANCP, Port Scan
	Remote System Discovery	2	ITL
	Serial Connection Enumeration	1	ITL
Lateral Movement	Exploitation of Remote Services	587	ITL, LANCP, Malformed Packets
	Remote File Copy	2	ITL
	Valid Accounts	35	ITL

Tactic	Technique	eyeInspect Event Types	eyeInspect Detection Engines
Collection	Data from Information Repositories	2	ITL
	Monitor Process State	8	ITL
	Point & Tag Identification	14	ITL, Malformed Packets
	Program Upload	16	ITL, Script
Command and Control	Commonly Used Port	3	ITL
	Connection Proxy	1	ITL
	Standard Application Layer Protocol	10	ITL, LANCP
Inhibit Response Function	Alarm Suppression	26	ITL, Script
	Block Command Message	2	ITL
	Block Serial COM	2	ITL
	Data Destruction	4	ITL
	Denial of Service	588	ITL, Malformed Packets
	Device Restart/Shutdown	14	ITL
	Manipulate I/O Image	18	ITL
	Modify Alarm Settings	4	ITL, Script
	Program Download	23	ITL, Script
	System Firmware	14	ITL
	Utilize/Change Operating Mode	2	ITL

Tactic	Technique	eyeInspect Event Types	eyeInspect Detection Engines
Impair Process Control	Change Program State	30	ITL
	Modify Control Logic	31	ITL
	Modify Parameter	14	ITL, DPBI
	Program Download	23	ITL, Script
	Rogue Master Device	4	LANCP, ITL
	Service Stop	4	ITL
	Unauthorized Command Message	17	ITL, DPBI, LANCP
Impact	Loss of Control	11	ITL
	Loss of Safety	2	ITL
	Loss of View	4	ITL
	Manipulation of Control	64	ITL, Script
	Manipulation of View	20	ITL

2.3. Evaluating Detection

Ideally, detection capabilities in a **SOC should be evaluated by using a red team that tries to cover as much as possible of the ATT&CK for ICS matrix** while a blue team responds to those attacks. This can uncover blind spots in detection, such as events that can't be observed with the current data sources or those that are observed but not flagged by the tools in place. Notice that detection in a real scenario depends not only on the tools that are used, but also on the placement of sensors in the network and their configuration.

To keep things simple, **we have showcased our threat detection and mapping capabilities using a dataset** with traffic captured from the DEF CON 27 ICS Village **Capture the Flag (CTF) competition**. The competition simulated an ICS network with a diversity of industrial devices such as an SEL-351 Protection System, a Schneider M221 PLC and a Siemens KTP400 HMI. The dataset includes traffic from standard protocols such as Modbus, BACnet, DNP3, Ethernet/IP and Profinet, as well as proprietary protocols such as SEL Fast Message and Siemens S7.

This dataset is interesting because it represents a realistic attack scenario where multiple threat actors are trying to reach different goals at the same time. It is also important to note that since this data comes from a competition environment, we won't see every possible attack tactic that would be observed by a cybersecurity stakeholder. For instance, the attackers were already in the network so there was no need for **Initial Access**. They weren't trying to hide their actions, so there was no need for **Evasion**. The goal of the CTF was not to disrupt any physical process, so there was no need for **Impact**.

The **detection results** for this dataset are shown in **Figure 8**, where detected techniques are highlighted in blue, and detailed in **Table 2**. Notice that we **detected** events related to **8 different tactics with just 4 of the detection engines**, spanning almost a whole attack lifecycle (the 3 exceptions are the ones mentioned above).

Initial Access	Execution	Persistence	Evasion	Discovery	Lateral Movement	Collection	Command and Control	Inhibit Response Function	Impair Process Control	Impact
Data Historian Compromise	Change Program State	Hooking	Exploitation for Evasion	**Control Device Identification**	Default Credentials	Automated Collection	Commonly Used Port	Activate Firmware Update Mode	Brute Force I/O	Damage to Property
Drive-by Compromise	Command-Line Interface	Module Firmware	Indicator Removal on Host	I/O Module Discovery	**Exploitation of Remote Services**	Data from Information Repositories	**Connection Proxy**	Alarm Suppression	Change Program State	Denial of Control
Engineering Workstation Compromise	Execution through API	Program Download	Masquerading	Network Connection Enumeration	External Remote Services	Detect Operating Mode	**Standard Application Layer Protocol**	Block Command Message	Masquerading	Denial of View
Exploit Public-Facing Application	Graphical User Interface	Project File Infection	Rogue Master Device	**Network Service Scanning**	Program Organization Units	Detect Program State		Block Reporting Message	Modify Control Logic	Loss of Availability
External Remote Services	**Man in the Middle**	System Firmware	Rootkit	Network Sniffing	Remote File Copy	I/O Image		Block Serial COM	**Modify Parameter**	Loss of Control
Internet Accessible Device	Program Organization Units	**Valid Accounts**	Spoof Reporting Message	Remote System Discovery	Valid Accounts	Location Identification		Data Destruction	Module Firmware	Loss of Productivity and Revenue
Replication Through Removable Media	Project File Infection		Utilize/Change Operating Mode	Serial Connection Enumeration		Monitor Process State		**Denial of Service**	Program Download	Loss of Safety
Spearphishing Attachment	Scripting					**Point & Tag Identification**		Device Restart/ Shutdown	Rogue Master Device	Loss of View
Supply Chain Compromise	User Execution					Program Upload		Manipulate I/O Image	**Service Stop**	Manipulation of Control
Wireless Compromise						Role Identification		Modify Alarm Settings	Spoof Reporting Message	Manipulation of View
						Screen Capture		Modify Control Logic	**Unauthorized Command Message**	Theft of Operational Information
								Program Download		
								Rootkit		
								System Firmware		
								Utilize/Change Operating Mode		

Figure 8: Techniques detected by eyeInspect from the DEF CON dataset.

Blue – Techniques detected by eyeInspect from the DEF CON dataset

Table 2: DEF CON Dataset Detailed Rresults

eyeInspect Event Type	eyeInspect Detection Engine	Tactic	Technique
ARP Poisoning	Man in the Middle	Execution	Man in the Middle
ICMP spoofed Redirect message	Man in the Middle	Execution	Man in the Middle
Login attempt using blacklisted credentials	ITL	Persistence	Valid Accounts
TCP NULL portscan	Port Scan	Discovery	Network Service Scanning
Distributed TCP SYN portscan	Port Scan	Discovery	Network Service Scanning
Modbus/TCP Read Device Identification command	ITL	Discovery	Control Device Identification
Modbus/TCP Report Slave ID command	ITL	Discovery	Control Device Identification
Invalid field length	Malformed Packets	Lateral Movement, Inhibit Response Function	Exploitation of Remote Services, Denial of Service
ICMP deprecated control message type	Malformed Packets	Lateral Movement, Inhibit Response Function	Exploitation of Remote Services, Denial of Service
ICMP invalid IP address in Redirect message	Malformed Packets	Lateral Movement, Inhibit Response Function	Exploitation of Remote Services, Denial of Service
IP duplicate fragment	Malformed Packets	Lateral Movement, Inhibit Response Function	Exploitation of Remote Services, Denial of Service
MODBUS invalid byte count in read coils function	Malformed Packets	Lateral Movement, Inhibit Response Function	Exploitation of Remote Services, Denial of Service
MODBUS invalid conformity level	Malformed Packets	Lateral Movement, Inhibit Response Function	Exploitation of Remote Services, Denial of Service
RPC/DCOM invalid version field value	Malformed Packets	Lateral Movement, Inhibit Response Function	Exploitation of Remote Services, Denial of Service
TCP invalid ACK number field value	Malformed Packets	Lateral Movement, Inhibit Response Function	Exploitation of Remote Services, Denial of Service
TCP invalid flags field value	Malformed Packets	Lateral Movement, Inhibit Response Function	Exploitation of Remote Services, Denial of Service
TCP invalid reserved field value	Malformed Packets	Lateral Movement, Inhibit Response Function	Exploitation of Remote Services, Denial of Service

eyeInspect Event Type	eyeInspect Detection Engine	Tactic	Technique
TCP flags values mismatch	Malformed Packets	Lateral Movement, Inhibit Response Function	Exploitation of Remote Services, Denial of Service
Illegal data address error from Modbus/TCP slave	ITL	Collection	Point and Tag Identification
Possible DNS tunneling attempt	ITL	Command and Control	Connection Proxy
Use of insecure SSL protocol version (SSLv3)	ITL	Command and Control	Standard Application Layer Protocol
Use of insecure protocol (TELNET)	ITL	Command and Control	Standard Application Layer Protocol
Blacklisted SSL client application	ITL	Command and Control	Standard Application Layer Protocol
Device with many failed connection attempts	ITL	Inhibit Response Function	Denial of Service
MODBUS/TCP device with unstable connection	ITL	Inhibit Response Function	Denial of Service
Host not receiving answers to DNS requests	ITL	Impair Process Control	Service Stop
Illegal data value error from Modbus/TCP slave	ITL	Impair Process Control	Modify Parameter
Illegal function error from Modbus/TCP slave	ITL	Impair Process Control	Unauthorized Command Message

3. Case Study: Detecting a Cyberattack

For illustrative purposes, we'll use a recreation of the **Stuxnet incident** to observe how eyeInspect behaves when encountering a real cyberattack. Although Stuxnet is a dated piece of malware, it's still **representative of the complexity of real targeted attacks** and became infamous for many reasons, among them:

- The use of four zero-day vulnerabilities
- Its ability to "cross air gaps" by infecting networks via USB flash drives
- The infection of different kinds of assets, such as computers running Windows, engineering workstations running industrial applications and Siemens S7 PLCs
- Its stealthy action when infecting computers that were not its final target

3.1. What a Real Incident Looks Like

Stuxnet is a worm that targets **Siemens S7 PLCs connected to Windows-based engineering workstations** [14]. The final goal of the malware was to disrupt the uranium enrichment process of nuclear research facilities by rapidly changing the speed of PLC-controlled centrifuges that separate nuclear material. However, besides reaching its target, **the worm also infected hundreds of thousands of computers in other organizations along its way** [15].

The operation of the malware can be **summarized**, at a very high-level, as follows [16]:

1. Stuxnet **enters the network via an infected USB stick**.
2. Stuxnet **searches for computers running industrial applications connected to Siemens S7 PLCs** in the network.
3. Stuxnet **tries to connect to a Command and Control server on the Internet** to update itself.
4. Stuxnet compromises and **modifies the logic running on the targeted PLCs**.
5. Stuxnet reports back **fake process control information to other controllers** so that they do not know the real state of the process.

These steps can be mapped to ATT&CK for ICS as shown in **Figure 9**. There are intermediate steps that we skip here for the sake of simplicity, but the full list of techniques used by Stuxnet can be found on [MITRE's website].

Step	1	2	3	4	5
Tactic	Initial Access	Discovery	Command and Control	Impair Process Control	Impact
Technique	Replication Through Removable Media	Remote System Discovery	Commonly Used Port	Program Download	Manipulation of View

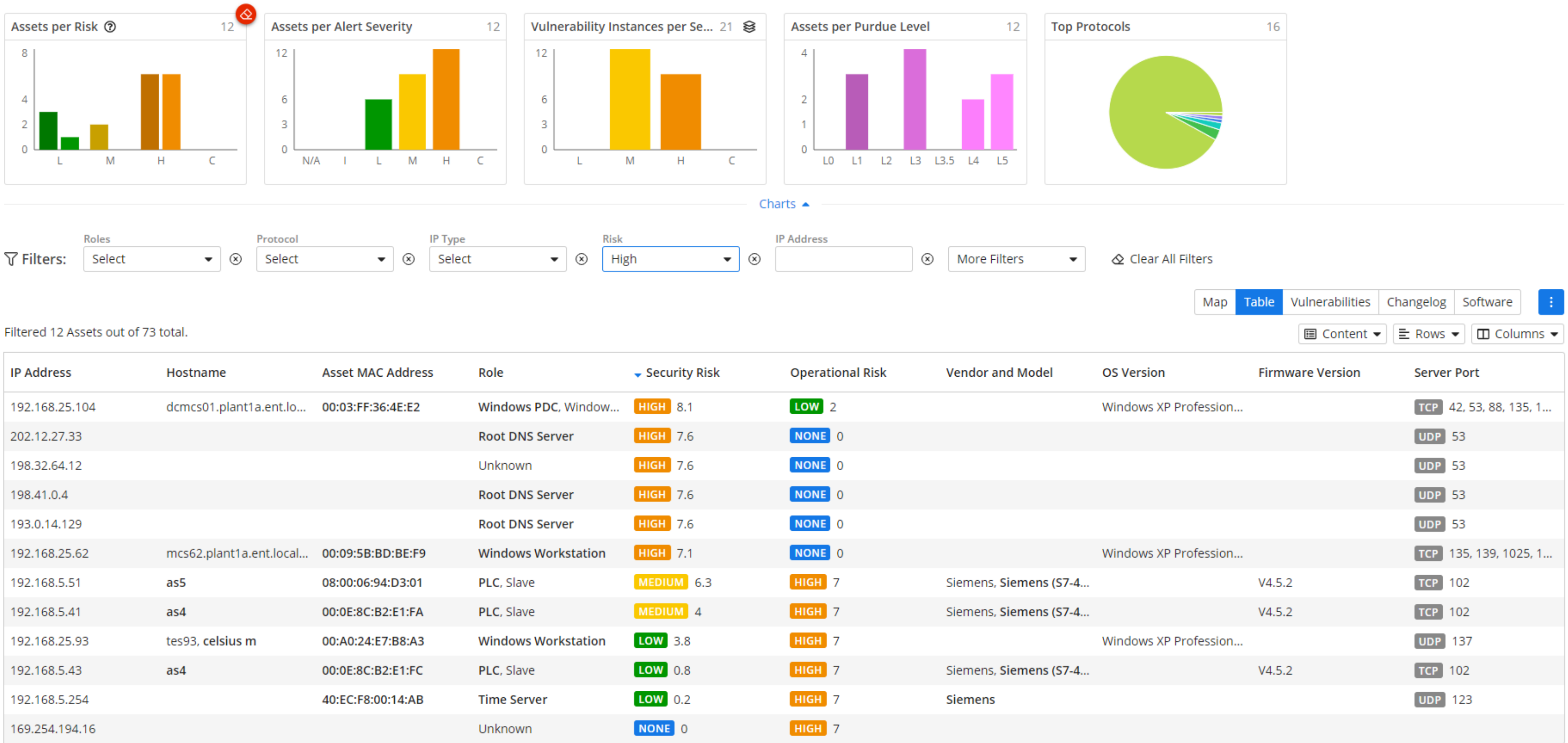
Figure 9: Stuxnet tactics and techniques.

To show how the Detection & Analysis phase of this incident would take place in a targeted industrial network, we replayed a traffic capture containing a sample of 30 minutes of infected traffic on eyeInspect. In just 30 minutes, the total number of identified assets was 73 and the total number of events observed by eyeInspect was 6,794.

So where does an analyst start the investigation as to what is going on in the network?

3.2. Investigating the Incident

Because eyeInspect **automatically calculates the risk of every asset identified in the network,** the analyst can **start the investigation not by looking at a specific event, but by looking into the riskiest assets** in the network, as shown **in Figure 10.** Notice that a **Windows Domain Controller is the asset with the highest risk score.**



IP Address	Hostname	Asset MAC Address	Role	Security Risk	Operational Risk	Vendor and Model	OS Version	Firmware Version	Server Port
192.168.25.104	dcmcs01.plant1a.ent.lo...	00:03:FF:36:4E:E2	Windows PDC, Window...	HIGH 8.1	LOW 2		Windows XP Profession...		TCP 42, 53, 88, 135, 1...
202.12.27.33			Root DNS Server	HIGH 7.6	NONE 0				UDP 53
198.32.64.12			Unknown	HIGH 7.6	NONE 0				UDP 53
198.41.0.4			Root DNS Server	HIGH 7.6	NONE 0				UDP 53
193.0.14.129			Root DNS Server	HIGH 7.6	NONE 0				UDP 53
192.168.25.62	mcs62.plant1a.ent.local...	00:09:5B:BD:BE:F9	Windows Workstation	HIGH 7.1	NONE 0		Windows XP Profession...		TCP 135, 139, 1025, 1...
192.168.5.51	as5	08:00:06:94:D3:01	PLC, Slave	MEDIUM 6.3	HIGH 7	Siemens, Siemens (S7-4...		V4.5.2	TCP 102
192.168.5.41	as4	00:0E:8C:B2:E1:FA	PLC, Slave	MEDIUM 4	HIGH 7	Siemens, Siemens (S7-4...		V4.5.2	TCP 102
192.168.25.93	tes93, celsius m	00:A0:24:E7:B8:A3	Windows Workstation	LOW 3.8	HIGH 7		Windows XP Profession...		UDP 137
192.168.5.43	as4	00:0E:8C:B2:E1:FC	PLC, Slave	LOW 0.8	HIGH 7	Siemens, Siemens (S7-4...		V4.5.2	TCP 102
192.168.5.254		40:EC:F8:00:14:AB	Time Server	LOW 0.2	HIGH 7	Siemens			UDP 123
169.254.194.16			Unknown	NONE 0	HIGH 7				

Figure 10: Assets with the highest risk score.

The analyst can then investigate **why this asset is the riskiest**. The answer is that there are 2,134 security-relevant events associated with it, as shown in **Figure 11**.

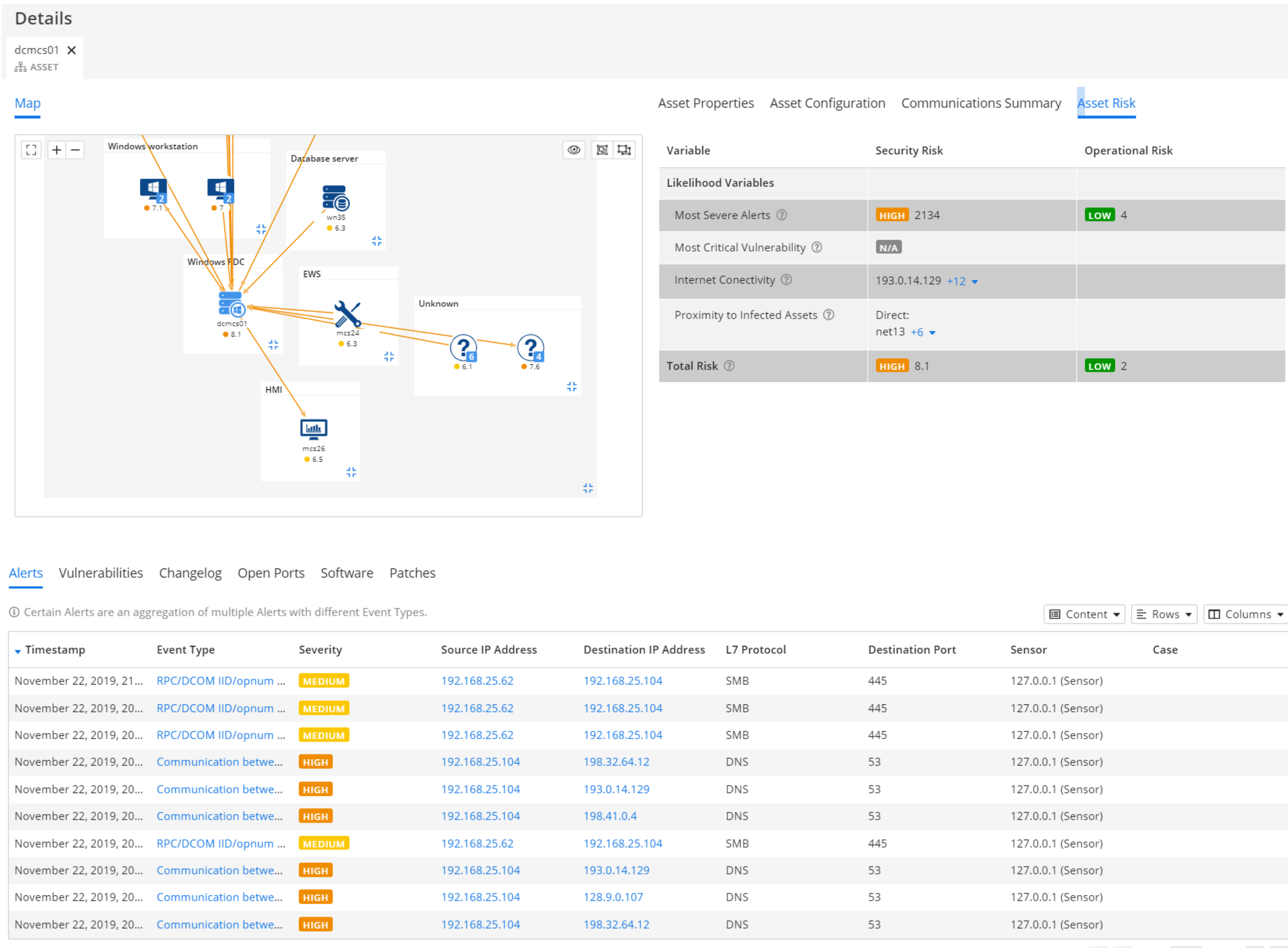


Figure 11: Risk components for the asset with the highest risk score.

For the alerts related to this asset, the analyst can then **visualize how they map to techniques in ATT&CK for ICS**, as shown in **Figure 12**. This provides a **"sequential" understanding** of events, not just with timestamps, but also defining **what stage of an attack they represent**.

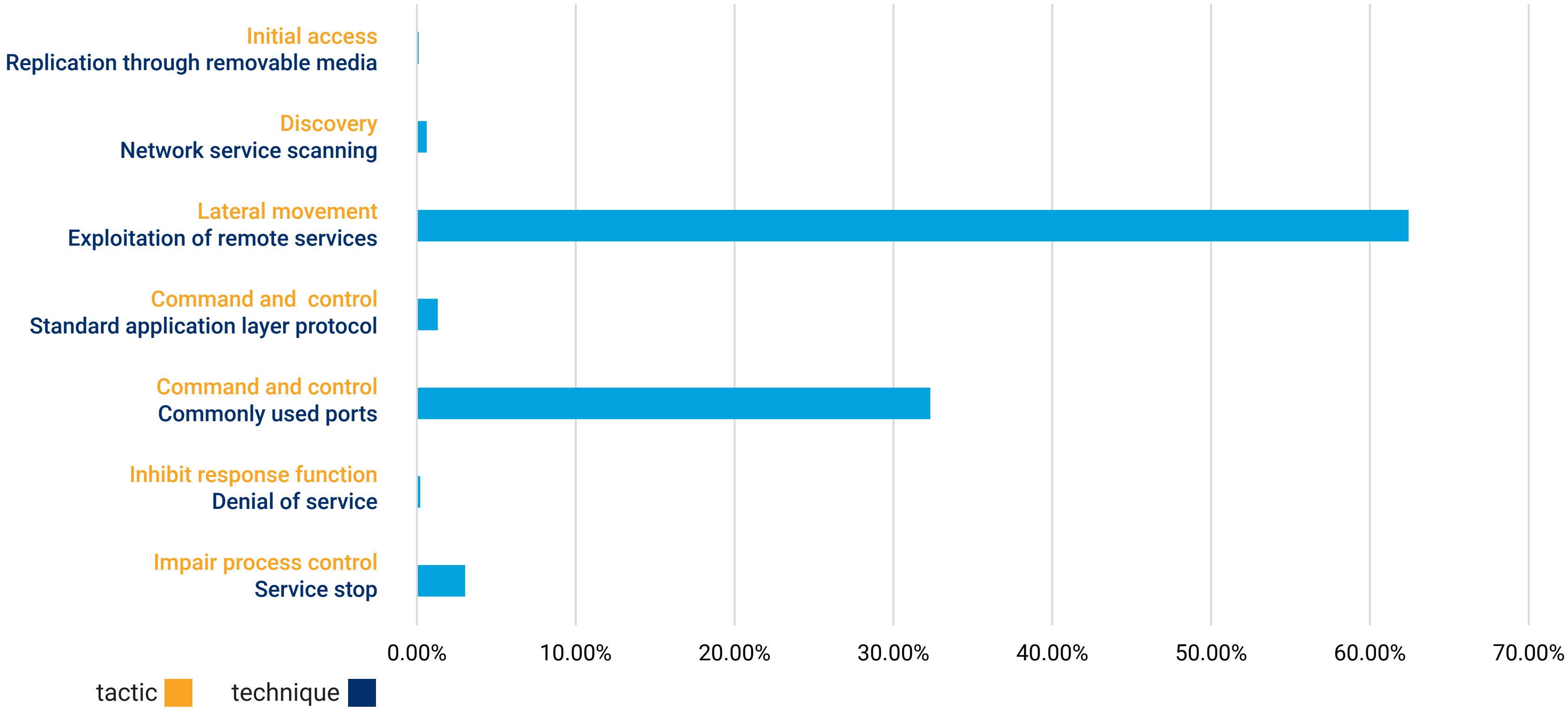


Figure 12: Sequential view of a Stuxnet incident by stage of attack mapped to ATT&CK for ICS.

Thus, the analyst can **start the investigation of alerts with those events mapped to techniques in the Initial Access tactic.**

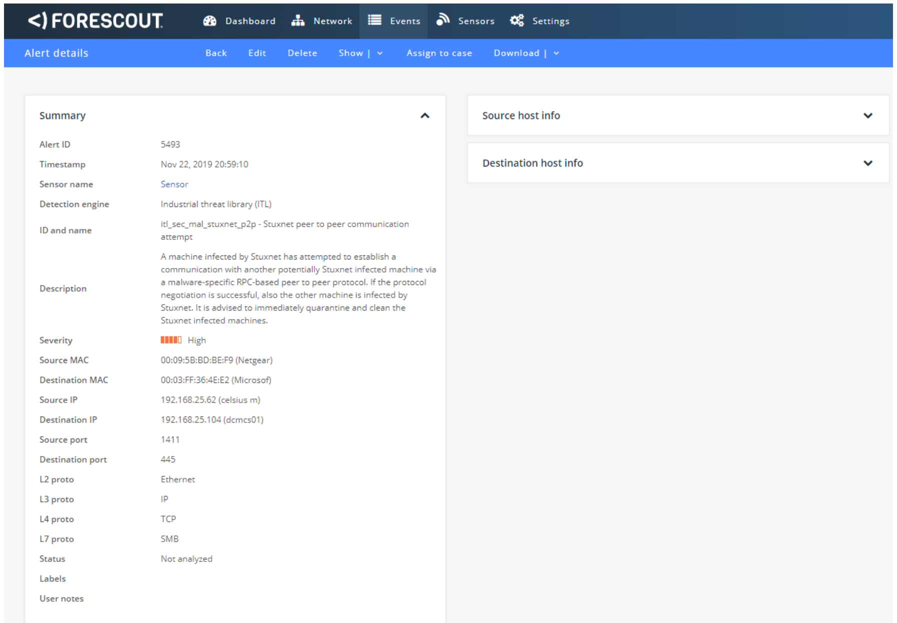


Figure 13: Replication through removable media.

The analyst will notice that eyeInspect detected an [Initial Access]() infection via [Replication Through Removable Media]() with the ITL checks for RPC-based peer-to-peer communication used by Stuxnet **(Figure 13)**.

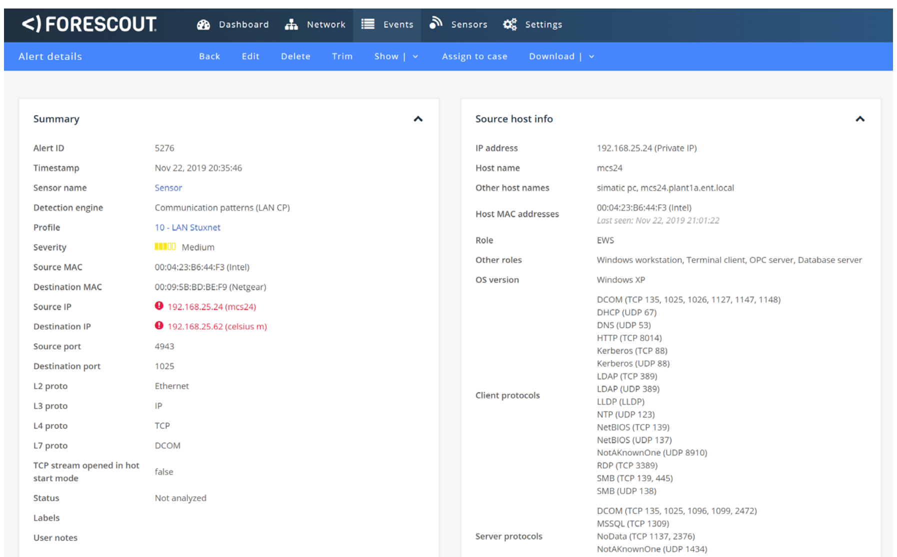


Figure 14: Remote system discovery.

From this step, the analyst can **follow the sequence of tactics in ATT&CK for ICS to continue the investigation**. At each step, this allows the analyst to filter out the alerts that are not relevant to the current stage of the attack being investigated.

The analyst would first see that eyeInspect detected Stuxnet's [Discovery]() of other devices in the network via [Remote System Discovery]() as an anomalous communication between two Windows XP machines **(Figure 14)**.

2

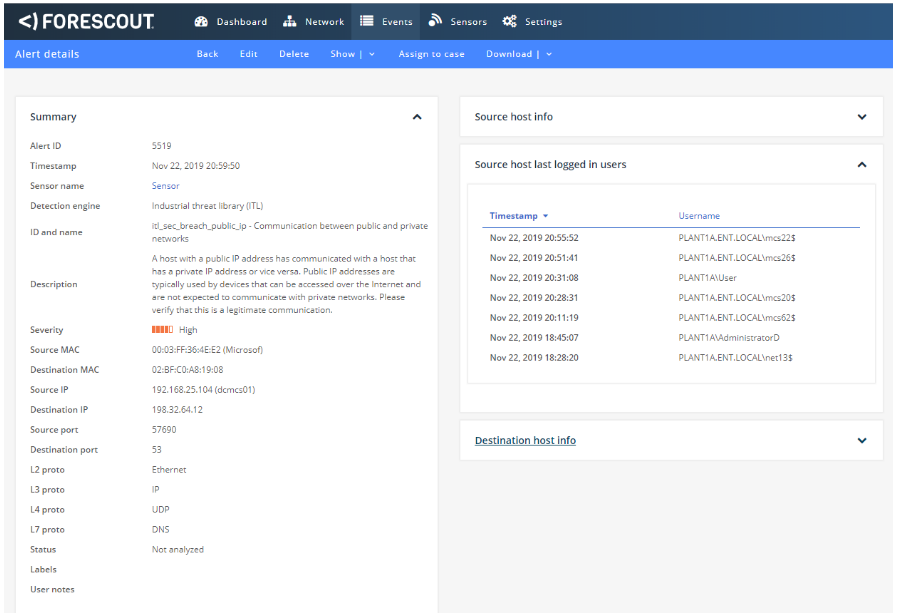


Figure 15: Commonly used port.

Second, eyeInspect detected the malware's communication to a Command and Control server and update attempt via a Commonly Used Port as anomalous communication between an internal host and a public IP address **(Figure 15)**.

3

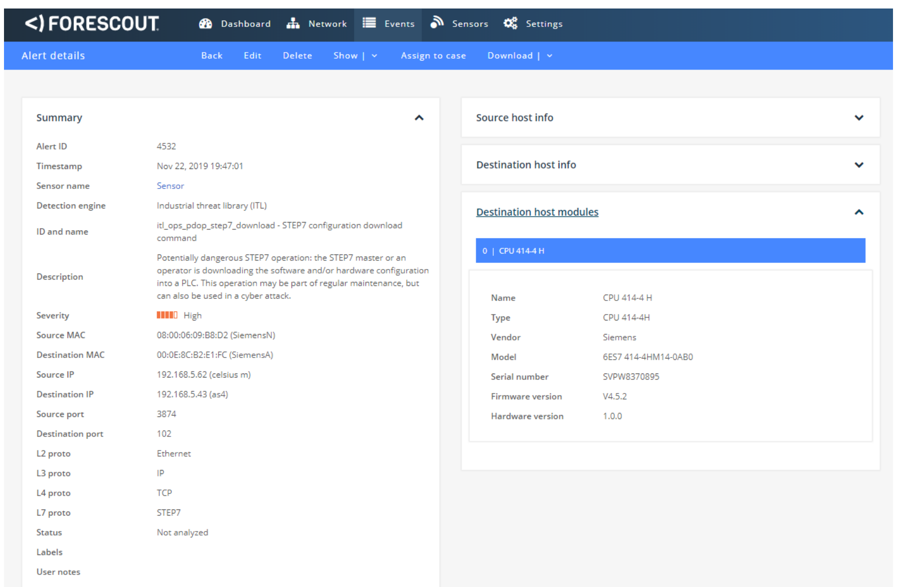


Figure 16: Program download.

Third, eyeInspect detected the compromise to Impair Process Control via Program Download of modified logic as a dangerous operation performed on the target PLC **(Figure 16)**.

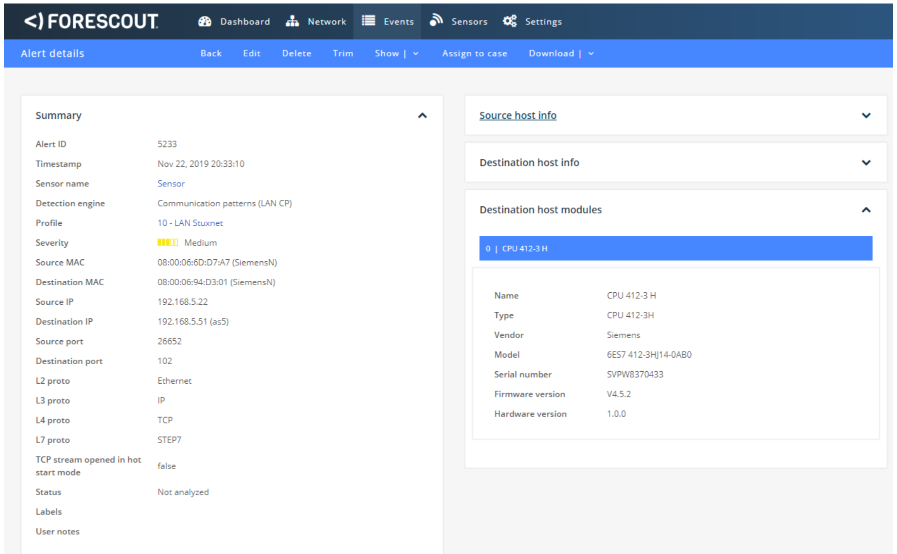


Figure 17: Manipulation of view.

Fourth, eyeInspect detected the Impact via Manipulation of View because of an anomalous communication pattern: writing variables in a different register **(Figure 17)**.

This concludes the investigation of the incident and, at this point, it's time to contain the damage being caused by the malware.

4. Improving Incident Containment, Eradication & Recovery

After the investigation in the Detection & Analysis phase, we need to **contain the incident, eradicate its presence in the network and recover from its damages.**

Eradication involves steps such as removing malware, deleting compromised accounts and patching vulnerabilities. Recovery involves restoring systems to their previous operational state, which may be accomplished by retrieving backups or sometimes rebuilding from scratch. **Our focus in this section**, however, **is the Containment step**, which is a prerequisite for the remediations applied in Eradication and Recovery.

Containment means stopping threats from further engaging in lateral movement within the network and causing greater damage. Containment should include isolating or shutting down systems, users and functions in the network, as well as redirecting attackers to sandbox systems. Of course, **devising an "appropriate response" is the challenge in this phase**, since it depends on the identified threat. **That said, containment is much easier to do when there are pre-determined response templates**. These templates, commonly known as **playbooks** [17], can guide analysts during investigation and response and allow orchestrated and automated execution of tasks by various security tools, using SOAR platforms.

Containment playbooks should be created for each type of incident and can be adapted from existing templates for the needs of the organization [18]. ATT&CK for ICS provides a common language and knowledgebase that can help organizations in tailoring playbooks to their needs and in assessing whether their existing playbooks can mitigate threats in realistic scenarios. One possible containment measure for the incident described in Section 3 is to block the engineering workstations used to compromise the PLCs from the network.

Below, we show an example playbook that could be used to implement these containment strategies. It leverages eyeInspect for detection and analysis, [Forescout eyeControl] for containment and the [Splunk Phantom SOAR] for orchestration.

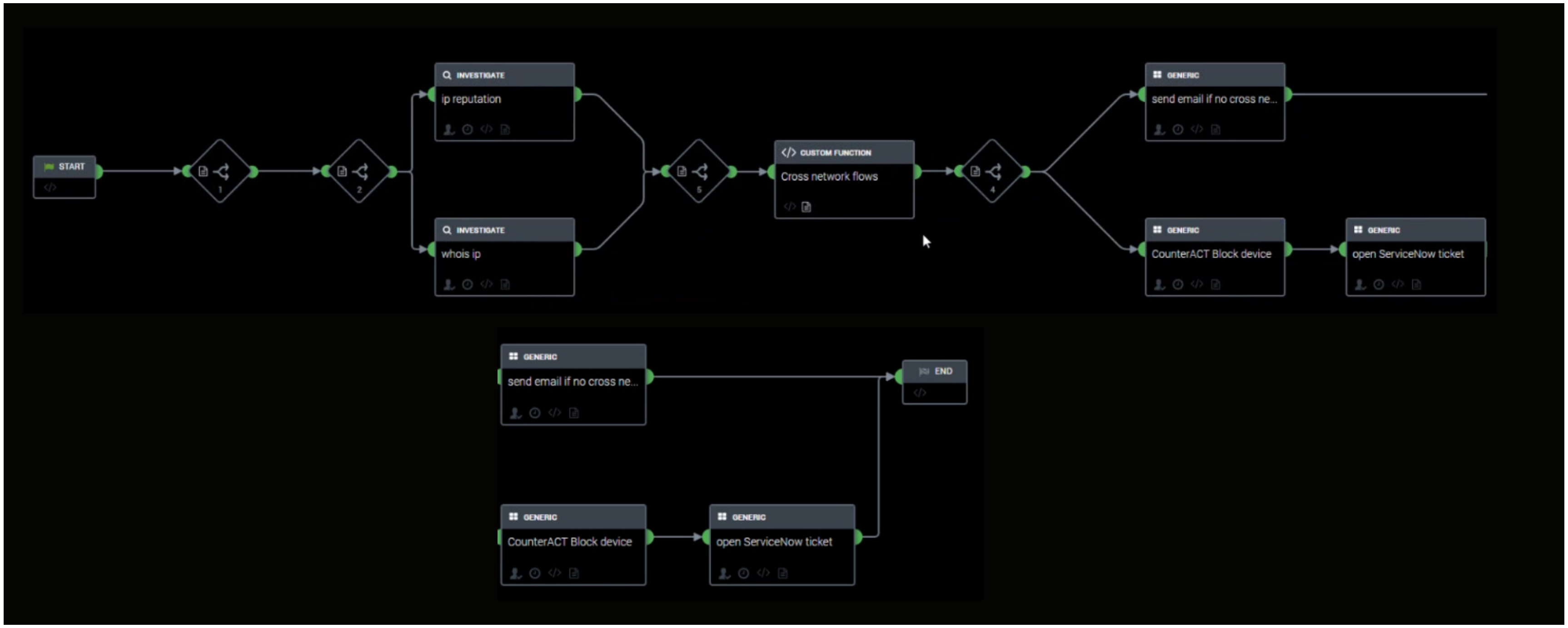


Figure 18: Orchestrated Response using the Phantom SOAR.

The playbook starts with the Stuxnet Command and Control communication detected by eyeInspect. The analyst can then investigate whether the IP addresses being contacted are malicious using reputation sources. If the communication is suspicious, the tool can check whether the involved asset has a cross-network flow, including whether it connects devices in different levels of the Purdue model, which can be a strong indicator of malicious activities at the operational layer. If this is happening, the device can be quarantined by Forescout eyeControl, and a ServiceNow ticket can be opened to inform IT staff of the situation.

This containment scenario highlights the integrations between eyeInspect, Splunk Phantom and eyeControl. It's just one example of the capabilities provided by the Forescout platform (shown in **Figure 19**) and its integrations via [eyeExtend].

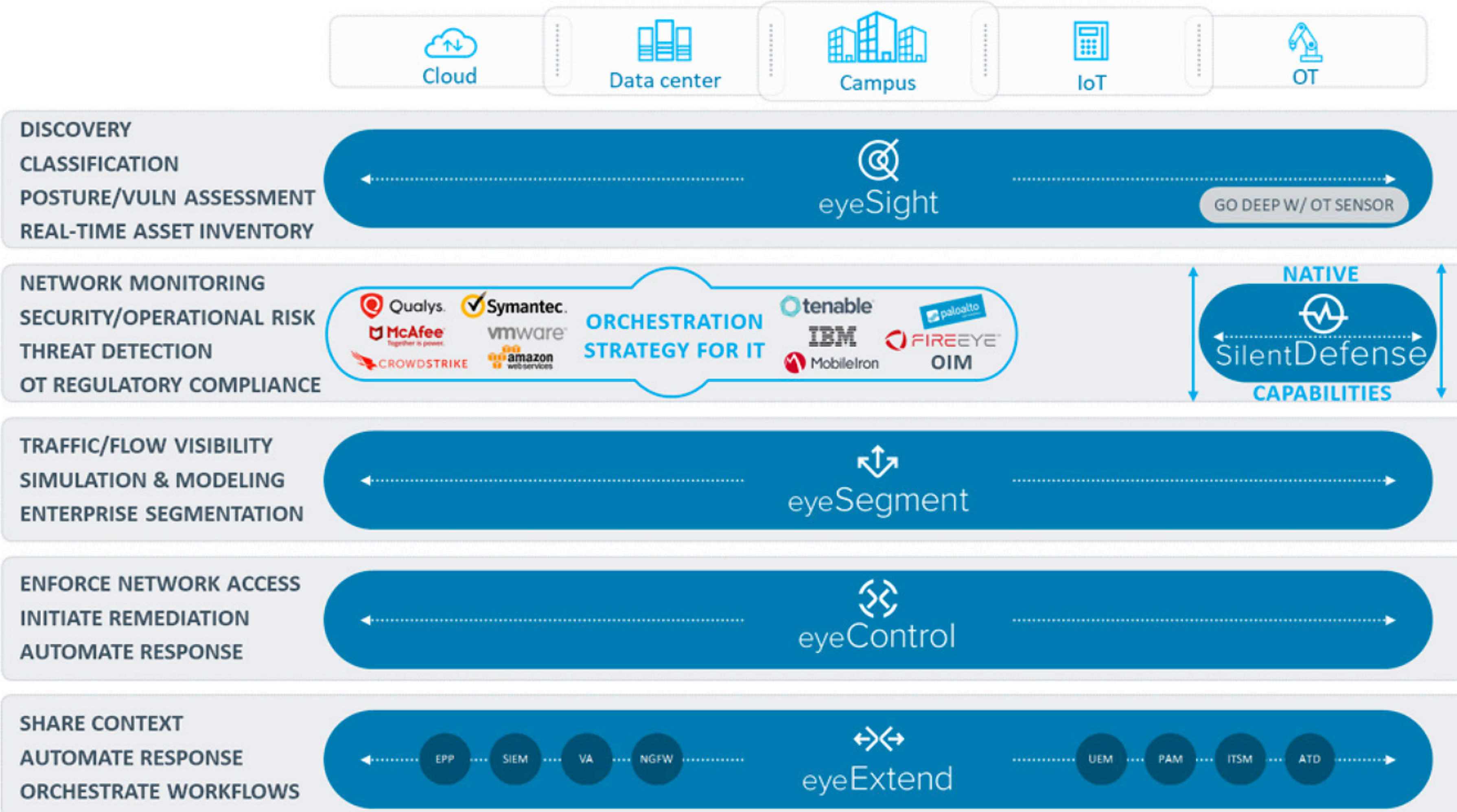


Figure 19: A representation of key capabilities of the Forescout platform.

5. Post-Incident: Conclusions

Adversarial frameworks such as MITRE ATT&CK for ICS are critical for bridging the semantic gap between attackers and defenders. Operationalizing these models into the incident response lifecycle using dedicated tools are worthwhile investments. As demonstrated in the ATT&CK for ICS framework use cases, eyeInspect detects many relevant OT-specific events, which empowers analysts with critical data to map those events to TTPs.

The key takeaways can be summarized as follows:

- ATT&CK for ICS helps **enhance existing OT-focused SOCs or helps to set up new SOCs** by providing a standardized set of TTPs to measure detection capabilities.
- Using eyeInspect + ATT&CK for ICS helps to **streamline incident response** by empowering analysts with an effective tool and procedure to quickly investigate incidents.
- Forescout enables a **holistic, OT-specific cybersecurity strategy** from detection to response by integrating with existing enterprise ecosystems to better orchestrate threat containment efforts.

More advanced uses of MITRE ATT&CK matrices that are not discussed in this paper include:

- **Threat hunting** to proactively look for the presence of threats using known TTPs in a network[19]. This is not a substitute for reactively detecting and responding to incidents. However, it can be an effective complement that helps to identify potential blind spots in the network. For more information on effective threat hunting in ICS networks, read this blog post.
- Developing effective **security controls** by mapping them to the techniques they can mitigate while seeing current gaps [20]. This allows organizations to fix their blind spots and improve their preparation for future incidents. For more information on planning effective security controls, read this blog post.
- Planning and automation of **adversary emulation** for penetration testing or red teaming [21]. This allows organizations to test their security strategy and even their incident response capabilities in realistic scenarios.

References

[1] D. Zafra, K. Lunden, N. Brubaker and J. Kennelly, "Ransomware Against the Machine: How Adversaries are Learning to Disrupt Industrial Production by Targeting IT and OT," 2020. [Online]. Available: https://www.fireeye.com/blog/threat-research/2020/02/ransomware-against-machine-learning-to-disrupt-industrial-production.html.

[2] R. Brown and R. Lee, "The Evolution of Cyber Threat Intelligence (CTI): 2019 SANS CTI Survey," 2019. [Online]. Available: https://www.sans.org/reading-room/whitepapers/threats/evolution-cyber-threat-intelligence-cti-2019-cti-survey-38790.

[3] S. Gianvecchio, C. Burkhalter, H. Lan, A. Sillers and K. Smith, "Closing the Gap with APTs Through Semantic Clusters and Automated Cybergames," in 15th EAI International Conference on Security and Privacy in Communication Systems (SecureComm), 2019.

[4] S. Milajerdi, R. Gjomemo, B. Eshete, R. Sekar and V. Venkatakrishnan, "HOLMES: Real-time APT Detection through Correlation of Suspicious Information Flows," in Proceedings of the 40th IEEE Symposium on Security and Privacy, 2019.

[5] J. Lambert, "Defenders think in lists. Attackers think in graphs. As long as this is true, attackers win.," 2015. [Online]. Available: https://blogs.technet.microsoft.com/johnla/2015/04/26/defenders-think-in-lists-attackers-think-in-graphs-as-long-as-this-is-true-attackers-win/.

[6] Lockheed Martin, "Cyber Kill Chain," [Online]. Available: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html.

[7] FireEye, "ThreatSpace," [Online]. Available: https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/ds-threatspace.pdf.

[8] M. Assante and R. Lee, "The Industrial Control System Cyber Kill Chain," 2015. [Online]. Available: https://www.sans.org/reading-room/whitepapers/ICS/industrial-control-system-cyber-kill-chain-36297.

[9] MITRE, "MITRE ATT&CK," [Online]. Available: https://attack.mitre.org/.

[10] MITRE, "MITRE Releases Framework for Cyber Attacks on Industrial Control Systems," 2020. [Online]. Available: https://www.mitre.org/news/press-releases/mitre-releases-framework-for-cyber-attacks-on-industrial-control-systems.

[11] P. Cichonski, T. Millar, T. Grance and K. Scarfone, "Computer Security Incident Handling Guide," NIST, 2012. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf.

[12] A. Applebaum, S. Johnson, M. Limiero and M. Smith, "Playbook Oriented Cyber Response," in National Cyber Summit (NCS), 2018.

[13] R. Pompon, "Security Liability in an 'Assume Breach' World," 2018. [Online]. Available: https://www.darkreading.com/partner-perspectives/f5/security-liability-in-an-assume-breach-world/a/d-id/1331100.

[14] R. Langer, "To Kill a Centrifuge: A Technical Analysis of What Stuxnet's Creators Tried to Achieve," 2013. [Online]. Available: https://www.langner.com/to-kill-a-centrifuge/.

[15] Kaspersky, "Stuxnet: Zero victims," 2014. [Online]. Available: https://securelist.com/stuxnet-zero-victims/67483/.

[16] D. Kushner, "The Real Story of Stuxnet," 2013. [Online]. Available: https://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet.

[17] J. Bollinger, B. Enright and M. Valites, Crafting the InfoSec Playbook, O'Reilly, 2015.

[18] Incident Response Consortium, "Playbooks Gallery," [Online]. Available: https://www.incidentresponse.com/playbooks/.

[19] P. Delgado, "Developing an Adaptive Threat Hunting Solution: The Elasticsearch Stack," 2018. [Online]. Available: https://uh-ir.tdl.org/handle/10657/3108.

[20] P. Langlois and J. Franklin, "Prioritizing ATT&CK Informed Defenses the CIS Way," 2019. [Online]. Available: https://www.slideshare.net/attackcon2018/mitre-attckcon-20-prioritizing-attck-informed-defenses-the-cis-way-philippe-langlois-verizon-and-joshua-franklin-cis.

[21] MITRE, "CALDERA," [Online]. Available: https://www.mitre.org/research/technology-transfer/open-source-software/caldera.

Learn more at Forescout.com

FORESCOUT

Forescout Technologies, Inc.
190 W Tasman Dr.
San Jose, CA 95134 USA

Toll-Free (US) 1-866-377-8771
Tel (Intl) +1-408-213-3191
Support +1-708-237-6591