T845: PROGRAM UPLOAD

PURPOSE

This Recipe, based upon use of the CyOTE methodology[1] (Figure 1), provides asset owners and operators (AOO) with general guidance for confirming suspicion of the Program Upload attack technique for the Collection tactic as defined by the MITRE ATT&CK® for Industrial Control Systems (ICS) framework[2,3] allowing them to be able to make informed business decisions based on collaborative analysis of the nature and context of the attack. This document also includes supplemental material with suggestions and recommendations for securing assets and improving detection capabilities. Additional information on this technique can be found in the *Program Upload (T845) Technique Detection Capability Sheet* for the Collection tactic.[4]

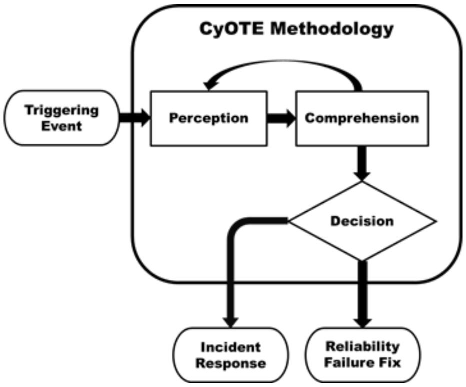

Figure 1: CyOTE Methodology Diagram

POTENTIAL ATTACK TARGETS

The program upload functionality is a common feature included in industrial protocols and is used by engineering workstations, human machine interfaces, and other industrial applications that manipulate programs on controllers. By leveraging the program upload functionality, an attacker can extract the program from a controller to the endpoint of their choosing and leverage the technique for industrial espionage or to assist with future phases of attack. Collecting and analyzing control program execution from targeted programmable logic controllers (PLC) devices provides adversaries information regarding processes controlled by the PLC, as well as potential vulnerabilities for exploitation.[5]

A capability for detecting the Program Upload technique requires AOOs to be able to parse and analyze several different communication protocols. Once identified, it is possible to extract the payload of the technique as it transits the network. Analysis of the uploaded PLC logic file allows the system operators to determine if the PLC program logic uploaded by the adversary has been modified previously. Consequently, the ability to parse and analyze network file transfer protocols and PLC communications enables the extraction of adversary payloads from the network stream.

[1] CESER, Methodology for Cybersecurity in Operational Technology Environments, CyOTE Program. Department of Energy, FY21, https://inl.gov/wp-content/uploads/2021/09/CyOTE-Methodology_2021.pdf.
[2] MITRE, Program Upload, 2021. Available online: https://collaborate.mitre.org/attackics/index.php/Technique/T0845.
[3] © 2021 The MITRE Corporation. This work is reproduced and distributed with the permission of The MITRE Corporation.
[4] CESER, Program Upload (T845) Technique Detection Capability Sheet, CyOTE Program. Department of Energy, FY21.
[5] MITRE, Program Upload, 2021. Available online: https://collaborate.mitre.org/attackics/index.php/Technique/T0845.

PERCEPTION: IDENTIFYING ANOMALIES

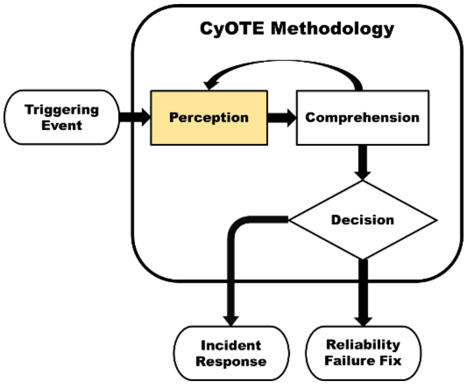

Figure 2: CyOTE Methodology – Perception Step

Perception (Figure 2) is the first active step in employing CyOTE's methodology. CyOTE uses the terms "perception" and "comprehension" as opposed to terms like "detection" and "understanding" for the same reasons that the North American Electric Reliability Corporation (NERC) uses those terms, which were adapted from Dr. Mica Endsley's model of situation awareness[6] – because they cause the reader to think of the necessary individual and organizational human cognition as opposed to merely automated data processing. Perception is defined as a signature capable of being detected by a human that was actually detected; perception does not mean opinion or subjective interpretation. It provides the starting point—detection of a triggering event in the organization—for investigation during the comprehension step. This stage involves the identification of observables, anomalies, and triggering events of a potential malicious event; please consult the CyOTE methodology for more information on these terms.[7]

EXAMPLE OBSERVABLES AND ANOMALIES OF THE PROGRAM UPLOAD TECHNIQUE

Differences between operational environments require AOOs to understand the generic and unique attributes of the environment under analysis. The anomalies and observables in Table 1 serve as a generic starting point to adapt to the realities of the specific environment. Each potential anomaly includes data sources where it may be observed, as well as what the observables may be.

Note that this table is not intended to be an exhaustive list of anomalies, data sources, or observables tied to the Program Upload technique.

Table 1: Notional Events

Observables	Anomalies	Data Sources
Large file or data transfers from an industrial device or usage of program upload Func codes from unexpected source host or subnet	Program upload requests coming from new or abnormal hosts or network segments	• Raw Network Data (Captured) • Raw Network Data (Live) • Network Flow Data (Captured) • Network Flow Data (Live)
Changes in statistical models or other patterns associated with program uploads	Program upload commands being sent at odd times or intervals	• Raw Network Data (Captured) • Raw Network Data (Live)

[6] Mica R. Endsley, "Situation Awareness Misconceptions and Misunderstandings," Journal of Cognitive Engineering and Decision Making 9, no. 1 (March 2015):4 https://doi.org/10.1177%2F1555343415572631.

[7] CESER, Methodology for Cybersecurity in Operational Technology Environments, CyOTE Program. Department of Energy, FY21, https://inl.gov/wp-content/uploads/2021/09/CyOTE-Methodology_2021.pdf.

Observables	Anomalies	Data Sources
Indications of a rogue application with a network socket or other logged behaviors to ports associated with program uploads might be present. It might only be possible to view signs of connection and not specific upload options.	An unexpected application on a system performs a program download	• Raw Network Data (Captured) • Raw Network Data (Live) • Windows Event Logs (Enhanced)
Unexpected changes to file metadata (file hash)	Unexpected program file modifications	• Raw Network Data (Captured) • Raw Network Data (Live) • Application Logs
• Device error messages indicating a device being in trouble • The inability to load a program into an engineering application might indicate a file is corrupt	Corrupt program file	• Raw Network Data (Captured) • Raw Network Data (Live)

STEPS FOR IDENTIFYING HIGH-CONSEQUENCE SYSTEMS, PATHWAYS, AND POTENTIAL ANOMALOUS PROGRAM UPLOADS

Asset owners and operators aiming to detect potential capabilities to monitor for use of the Program Upload technique should consider a phased approach to development to include continuous testing and evaluation throughout its life cycle. To complement this, it is highly encouraged to use the following steps to map out existing OT infrastructure both logically and physically. This supports capability development and the analysis of potential alerts, enabling the quick identification of control devices communicating within the infrastructure.

Continual testing and evaluation will ensure the newly introduced software does not negatively impact, adversely affect, or introduce vulnerabilities into the existing OT environment. As a guideline, during the development phase, secure coding practices should be employed.[8]

1. Identify what devices and protocols to monitor for program uploads
   a. E.g., remote terminal units (RTU)/automation controllers, PLC
   b. Identify parsers for the applicable protocols of each potential trigger
2. Identify the capability location and when it will operate
   a. Example capability locations: from firewall, integrated host, server, IDS/IPS, other
   b. Example operating timeframes: at startup, real-time, daily, weekly
3. Identify tap points (sensors) for observing device traffic
   a. This may include servers, switches, security appliances, and logging locations (hosts)

[8] Microsoft, "Security engineering SDL practices," Blog, available online at https://www.microsoft.com/en-us/securityengineering/sdl/practices.

      i. Plan sensor placement based on locations within the architecture that provide context related to the anomaly and prioritize systems at greater risk

b. Monitoring and traffic aggregation may necessitate tap placement on both sides of the identified devices
      i. E.g., MAC addresses may change as information traverses networking infrastructure like protocol converters

c. Recommend establishing capture requirements for monitoring OT traffic and their locations[9, 10]
      i. Storage (how much and for how long)
      ii. Line rate (e.g., 1/10/40/100 Gb)
      iii. Live stream data or full Packet Capture (PCAP) offline
      iv. Central versus distributed collection/analysis/alerting

COMPREHENSION: ANALYZING TRIGGERS AND THEIR CONTEXT

The second step of employing the CyOTE methodology is Comprehension (Figure 3). Comprehension is the ability to understand an anomaly in all its relevant context across the operations, OT, information technology (IT), business, and cybersecurity domains. Comprehension involves understanding the nature and possible origins of the anomaly and developing broader awareness of the overall context in which the triggering event came to be. Because of the multidisciplinary approach used for a sufficient investigation, comprehension for the purposes of CyOTE is an organizational ability, not an individual one.

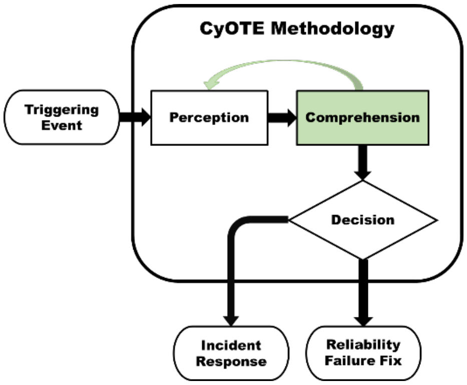

Figure 3: CyOTE Methodology - Comprehension Step

IDENTIFY AND COLLECT AVAILABLE INFORMATION RELATING TO PROGRAM UPLOADS

Developing comprehension around an anomaly is most effective and efficient when small core teams of full-time system operators, OT technicians, and cybersecurity analysts from different departments and external resources come together to purposefully focus on the problem in the context of their shared organization. Table 2 includes a list of different organizations that could be leveraged to collect information on an anomaly. Depending on the size and resources of your organization, some of these resources might not be applicable.

[9] CESER, Security Monitoring Best Practices, CyOTE, 2021.
[10] CESER, Lessons Learned, CyOTE Program, Department of Energy, 2021.

Table 2: Business Organizations that Support Information Collection for Program Uploads

Organization	Capacity
• System Operations Departments • Engineering Departments	Control center field operators and real-time engineers should be one of the first sources consulted. Information collected might include manual logs, notes from field personnel investigations, or other records if the anomaly involved a disruption in the system above some established threshold.
Cybersecurity Departments	Includes those responsible for the confidentiality, integrity, and availability of the organization's digital assets provide the threat-informed perspective and bring experience and capabilities to analyze situations and data for security issues.
Original Equipment Manufacturers (OEM)	Includes those who produce and support the hardware and software present within the industrial environment. OEMs might or might not be under support contracts but might provide technical documentation and expert advice on expected device behavior.
Third-Party Support Vendors	Cybersecurity, threat intel, and other vendors that provide subject matter expertise might be able to provide insight into anomalies surrounding the trigger conditions.

Since access to raw data typically requires coordination with human organizational oversight, it generally is better to pursue information and context from different departments within the organization. When needed, have them provide the identified data under their control for shared analysis. Appendix A provides specific datasets and information that could be helpful depending on the situation.

STEPS FOR IDENTIFYING AND EXTRACTING INFORMATION FROM OT NETWORK TRAFFIC FOR ANALYSIS OF PROGRAM UPLOADS

The information on high-consequence systems, pathways, and potential anomalies collected previously becomes more refined for collection in this process. Extracted information should be prioritized based on importance, with timelines established for capturing and holding information for analysis and review.

Suggested data fields to collect include:

- Timestamp
- Device Identifier (will vary based on environment)
  - Source and destination IP addresses
  - MAC addresses
- Program upload message
- Program payload

- Payload size (e.g., bytes)

STEPS FOR ANALYZING EXTRACTED FIELDS AND IDENTIFYING FIELD-LEVEL ANOMALIES FOR PROGRAM UPLOADS

The suggested fields above are applied to data analysis and used to help establish anomalies. The list below includes suggested ways to use the extracted fields to help identify anomalies that could be alerted on. Any message revealing two or more parameters are given greater precedence for analysis and correlation with other observables to identify potentially anomalous/malicious downloads.

1. Identify uploads of interest to alert on
    a. Document anomalies based on host
        i. Identify the existing traffic origination points
        ii. Include the frequency and type of upload(s)
    b. Identify and match uploads to high-risk devices with criticality to the physical process
        i. Determine if the alert is valid or invalid based on analysis of the message parameters and source(s)
2. Identify uploads coming from new or abnormal hosts
    a. Analyze host lists for uploads issued to end device(s)
    b. Conduct a comparative analysis to identify new connections and alerts versus older ones
    c. Determine whether uploads are occurring at an abnormal frequency
        i. E.g., frequency, order, type, messaging timing
        ii. Track uploads and perform statistical and/or procedural tests
3. Establish anomalies
    a. Incorporate the analysis findings provided earlier and implement to refine alert parameters to focus on the useful information and minimize the non-useful alerts
        i. E.g., new or abnormal uploads, high-risk device uploads

REPORTING TRIGGERING EVENTS

Security needs to consider the operational and business aspects of the organization. Timely reporting of anomalous activity reduces the time needed to comprehend triggering events and make a decision, leading to reduced process outage times. The following guidance aims to enhance your organization’s existing reporting procedures, although these guidelines might also vary based on the composition of your organization. If your industry or organization requires stricter timeframes or processes, you should default to the more stringent timeframe and procedures.

Programmatic alarms should already be logging information, but organizations should document any contextual information for further analysis and reporting. Organizations should also report unusual operating conditions not based on programmatic alarms. Examples of essential details to record include the usual and intended conditions and how the observed operating condition deviates from expected. Table 3 below provides reporting guidelines for a sample of triggering events but is by no means an exhaustive list.

Table 3: Triggering Event Reporting Suggestions for Program Uploads

What To Report	Whom To Report To	Recommended Timeframe	Desired Outcome
Unexpected and/or unauthorized access or traffic to destination device by source device	Team responsible for source and destination device	1 hour	Collaborative troubleshooting should occur to identify and remediate the root cause
Corrupt program file or integrity breach into OT system	Team responsible for managing OT assets	1 hour	● Collaborative troubleshooting should occur to identify and remediate the root cause ● The OT team knows that a potential intrusion might have occurred

ANALYZE INFORMATION AND CONTEXT COLLABORATIVELY

Documentation and Knowledge Management Process

Begin a documentation and knowledge management process in support of the investigation. Start with a determination of perceived actions in the triggering event. Did the change occur within:

- The physical domain (something involving telemetered quantities such as noise, voltage, current, frequency, or temperature, or the physical configuration of a piece of infrastructure); or
- The OT domain (something involving traffic or signals transiting a communication medium, or the logical configuration of a piece of infrastructure, or something involving memory/storage metrics); or
- Both the physical and the OT domains?

Given the determination of a physical or OT starting point, identify what expected corresponding, perceptible observables would exist in the other domain and search for their presence or absence. For example, an unplanned file server reboot initiated by a program that was downloaded might produce digital footprints like logs and errors. It might correspond to data loss, unauthorized configuration changes, elevated user permissions, or account activity at unusual hours.

Technical Analysis

The end goal for technical analysis should validate or disprove the hypotheses associated with the anomalies. Technical analysis on program uploads focuses on the communications between hosts and the program file content and metadata. Network analysis focused on known ports with unknown protocols might leverage large communication transfers to identify program uploads. For known protocols, program upload activity might focus on associated function codes

Technical analysis might also focus on the program file. Program file metadata (e.g., file hash) provides a high-level method to validate program integrity. Specific to uploads, validation of the integrity of a program transfer confirms the success of the transfer.

Context Building Questions

Network data, if available, can provide the initial cluster of devices to focus on for initial context analysis. The viability of network analysis might vary depending on enabled protocols and the ability to parse communications from the protocols. The following questions intend to assist with initial analysis:

- What protocols support program uploads in the environment? It is critical to identify all possible protocols to ensure adequate analytic coverage.
- What hosts were observed leveraging the previously discovered protocols? Do any of the observed communications include specific program upload commands or large file transfers?
- Does the observed behavior correlate with expected behaviors? Can the actions observed on the network or host be correlated with known operator action?

The existence of a program upload itself isn’t suspicious; however, unusual or unexpected program uploads might indicate nefarious activity. Further context development questions should focus on splitting known good behavior from any deviations from this known good.

Visual Information Modeling

Enumerate all the lines of questioning identified through this stage of the analysis of the anomaly. At this point, it is beneficial to begin a node and link diagram from the information documented in the knowledge management processes to help visualize relationships between observables; the CyOTE program colloquially refers to this observable linking diagram as a “worm diagram.”

The triggering event is the first node, with all its related observables radially connected to it; include both confirmed observables and observables that were expected but not found, with some sort of visual discriminator between presence and absence (e.g., solid or dashed lines). Highlight the triggering event if the event relates to the implementation of a specific adversary technique. Include links emanating from the triggering event representing the as-yet-unanswered questions considered, as well as links and additional nodes for answered questions that confidently satisfy the extent of a particular line of inquiry. A notional example of this diagram for an investigation in progress is shown in Figure 4.

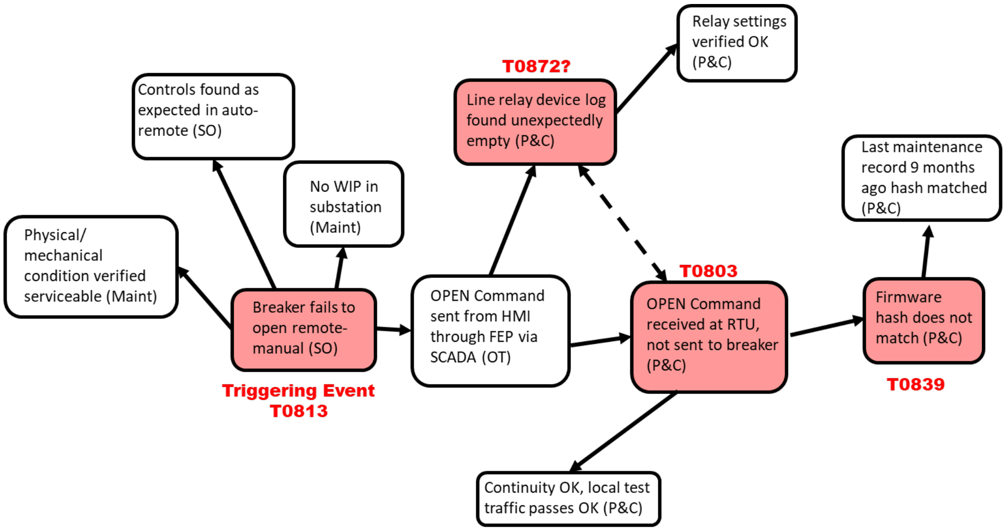
Relay settings verified OK (P&C)
T0872?
Line relay device log found unexpectedly empty (P&C)
Controls found as expected in auto-remote (SO)
No WIP in substation (Maint)
Last maintenance record 9 months ago hash matched (P&C)
Physical/ mechanical condition verified serviceable (Maint)
Breaker fails to open remote-manual (SO)
OPEN Command sent from HMI through FEP via SCADA (OT)
T0803
OPEN Command received at RTU, not sent to breaker (P&C)
Firmware hash does not match (P&C)
T0839
Triggering Event
T0813
Continuity OK, local test traffic passes OK (P&C)

Figure 4: Example CyOTE Observables Link Diagram

INVESTIGATE POTENTIALLY RELATED ANOMALIES TO PROGRAM UPLOADS

After an anomaly has been comprehended sufficiently and mapped to a technique, repeat the steps above, starting from each questioning line resulting from the initial anomaly analysis. The importance of recording and organizing the information discovered in the comprehension process and visualizing it through a node and link diagram becomes exponentially more important as the first perceived anomaly expands into a web of postulated, confirmed, and denied relationships.

DECISION: REMEDIATING THE CAUSE AND EFFECTS OF A TRIGGERING EVENT

The last step of the CyOTE methodology is the Decision step (Figure 5). In this step, the AOO makes a risk-informed business decision on how to proceed based on the information gathered in the previous steps. The level of comprehension and detail needed to make the decision will vary from company to company and may be related to resource availability. The length and consistency of a discovered and comprehended "worm diagram" representing a prospective kill chain fragment needed to decide to proceed with incident response will also vary based on a company's risk tolerance.

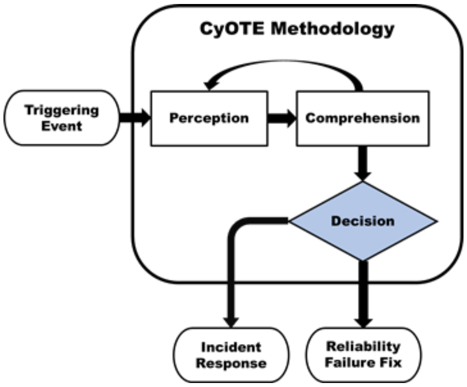

Figure 5: CyOTE Methodology - Decision Step

INCIDENT RESPONSE

In situations where there is sufficient belief that the anomalies perceived and comprehended indicate possible malicious cyber activity, the appropriate organizational action is to initiate a cybersecurity incident response process according to established policy and procedures. The information and context developed through the application of the CyOTE methodology will be helpful to incident handlers for developing and implementing appropriate mitigating actions. Consult with your organization’s incident response procedures for the next steps.

CORRECTIVE MAINTENANCE

When analysis fails to establish a reasonable indication of malicious cyber activity, the next step is to determine if the anomalies indicate that a non-malicious failure is occurring. One appropriate action involves resolving any deficiencies discovered through corrective maintenance and work management processes according to organizational policy and procedures. It is worthwhile to maintain records of these situations for future reference and comparison to subsequent anomalies. Consult with your organization’s engineers, operators, and other parties responsible for maintenance in your environment. For vendor-supported devices, coordinate with external parties when necessary.

IMPROVING SECURITY CAPABILITIES

Prevention and preparation include proactive measures to reduce the risk or likelihood of an attack or enable an organization to respond more rapidly. Preventive actions may raise the cost of an attack, which could deter the attacker from an act or eliminate the possibility that specific attack paths would be used altogether. AOOs should develop and deploy data collection management strategies to prepare for the collection of data needed to detect this technique and properly analyze potentially anomalous triggers.

CONTROL MATRIX FOR PROGRAM UPLOADS

The control matrix (Table 4) is a technical capability or system designed to prevent or sabotage specific attack techniques used by malicious actors. The control matrix assists with the identification of detection capability improvement areas.

Table 4: Control Matrix

Control	Matrix	Relevance
Network Segmentation	MITRE ATT&CK® for ICS: M0930[11]	When used in conjunction, network segmentation, network traffic community deviation, protocol metadata anomaly

[11] MITRE, Network Segmentation, 2021. Available online: https://collaborate.mitre.org/attackics/index.php/Mitigation/M0930.

Control	Matrix	Relevance
Network Traffic Community Deviation	MITRE D3FEND: D3-NTCD[12]	detection, and client-server payload profiling provide an exhaustive view into the network behaviors associated with the analysis of service stop commands. This group of controls does not exclusively include network data—the group can also include host-based application and operating system logs associated with the network traffic. ● Understand the statistical baseline for network traffic within your environment. This baseline might provide the opportunity to detect malicious program downloads. ● The baseline deviation might be calculated on the high-level communications summary or the specific function code or operation within a given protocol associated with program downloads.
Protocol Metadata Anomaly Detection	MITRE D3FEND: D3-PMAD[13]	
Client-Server Payload Profiling	MITRE D3FEND: D3-CSPP[14]	
Operating System Monitoring	MITRE D3FEND: D3-OSM[15]	Operating system monitoring focused on running processes and associated network ports can assist with identification and context analysis of potential nefarious uploads. ● Monitor for changes to the operating system's network stack. An attacker might install a driver filter to manipulate program upload requests
File Carving	MITRE D3FEND: D3-FC[16]	File carving assists with the investigation and confirmation of a program upload between two endpoints. ● File carving enables deeper network-based detection of programs transiting the network ● For completeness, you will need to ensure your file carving covers all possible protocols a controller might use for program uploads
Process Spawn Analysis	MITRE D3FEND: D3-PSA[17]	Process spawn and lineage analysis of applications responsible for program downloads assists with the determination of if a program upload is malicious or explainable. ● Process data might come from endpoint monitoring software or live analysis tools. ● Identify the application performing the program upload and leverage operating system event log and/or connection data from memory to identify the process responsible
Process Lineage Analysis	MITRE D3FEND: D3-PLA[18]	

[12] MITRE, Network Traffic Community Deviation, 2021. Available online: https://d3fend.mitre.org/technique/d3f:NetworkTrafficCommunityDeviation.

[13] MITRE, Protocol Metadata Anomaly Detection, 2021. Available online: https://d3fend.mitre.org/technique/d3f:ProtocolMetadataAnomalyDetection.

[14] MITRE, Client-server Payload Profiling, 2021. Available online: https://d3fend.mitre.org/technique/d3f:Client-serverPayloadProfiling.

[15] MITRE, Operating System Monitoring, 2021. Available online: https://d3fend.mitre.org/technique/d3f:OperatingSystemMonitoring.

[16] MITRE, File Carving, 2021. Available online: https://d3fend.mitre.org/technique/d3f:FileCarving.

[17] MITRE, Process Spawn Analysis, 2021. Available online: https://d3fend.mitre.org/technique/d3f:ProcessSpawnAnalysis.

[18] MITRE, Process Lineage Analysis, 2021. Available online: https://d3fend.mitre.org/technique/d3f:ProcessLineageAnalysis.

Control	Matrix	Relevance
Network Segmentation	ICS ATT&CK: M0930[19]	Network segmentation divides your network into sections based on manufacturer specification, role, or by the design of your organization. Network traffic filtering applies a set of rules at different points in the network or on a host to stop the communication of packets that meet a given signature. Network allowlists can also be used to block traffic based on traffic metadata such as IP address, ports, time, or other fields within a given communication stream. ● Program uploads should be limited to the network segments applicable to users and systems that need to perform the upload operation ● Device specific filtering between engineering workstation and other endpoints that should perform program uploads and the devices that the users can reach might also be filtered at the network and host level.
	ID: D3-NI (Network Isolation)[20]	
Filter Network Traffic	ICS ATT&CK: M0937[21]	
	ID: D3-ITF (Inbound Traffic Filtering)[22]	
	ID: D3-OTF (Outbound Traffic Filtering)[23]	
Network Allowlist	ICS ATT&CK: M0807[24]	
	ID: D3-DNSAL (DNS Allowlisting)[25]	

TESTING, DEPLOYING, AND REFINING ALERTING PARAMETERS FOR PROGRAM UPLOADS

The parameters and established anomalies from previous analysis lead to the continual testing and gradual deployment of the capability and evaluation of alerts. The resulting output functions and considerations are realized during this process, the products of which are sent to output destination(s) in selected format(s).

1. Validate triggers and alerts
   a. Ensure the capability does not conflict with existing monitoring functionality
   b. Ensure the capability does not adversely impact the existing environment
   c. Test alerting functions
      i. Use synthetic data (e.g., PCAPs containing program uploads)
      ii. If the test fails, re-evaluate the steps taken iteratively (line by line)
      iii. If successful, enact a graduated deployment schedule and retest for each iteration

[19] MITRE, Network Segmentation, 2021. Available online: https://collaborate.mitre.org/attackics/index.php/Mitigation/M0930.
[20] MITRE, Network Isolation, 2021. Available online: https://d3fend.mitre.org/technique/d3f:NetworkIsolation.
[21] MITRE, Filter Network Traffic, 2021. Available online: https://collaborate.mitre.org/attackics/index.php/Mitigation/M0937.
[22] MITRE, Inbound Traffic Filtering, 2021. Available online: https://d3fend.mitre.org/technique/d3f:InboundTrafficFiltering.
[23] MITRE, Outbound Traffic Filtering, 2021. Available online: https://d3fend.mitre.org/technique/d3f:OutboundTrafficFiltering.
[24] MITRE, Network Allowlists, 2021. Available online: https://collaborate.mitre.org/attackics/index.php/Mitigation/M0807.
[25] MITRE, DNS Allowlisting, 2021. Available online: https://d3fend.mitre.org/technique/d3f:DNSAllowlisting.

   d. Consider communication criteria for multiple locations and information consolidation during graduated deployment
2. Identify output destination(s) (SIEM, Splunk, Gravwell, Elk)
   a. Identify output format(s) (STIX, Syslog, JSON, CSV)
   b. Define actionable data requirements, processes, and responses
      i. Logging
      ii. Alert content
      iii. Alert response(s) (local or SOC)
3. Identify what information to log (long-term/short-term)
   a. The aggregation of different log types may assist in identifying potentially anomalous behaviors within OT environments

The overall output of this process may result in one of the following: script, application, Yara rule, SIEM plugin, or another tool capable of identifying and alerting on the use of the Program Upload technique within OT environments.

CONCLUSION

By employing the strategies in each step of the CyOTE methodology outlined in this Recipe, AOOs will be able to develop a capability to read and analyze their OT traffic and determine whether the Program Upload technique is being used as part of a malicious cyber-attack.

Through the Perception step, the AOO identifies observables and anomalies indicative of the Program Upload technique. Determining the associated observables and potential data sources for these anomalies will give the AOO the basis they need to begin comprehending the context in which a triggering event involving the Program Upload technique came to be. Large, unexpected file or data transfers; abnormal program upload patterns; unexpected changes to file metadata; rogue applications; and device corruption are all potential observables that could indicate the use of Program Upload technique. Anomalies tied to these observables could be corrupted program files, unexpected program file modifications, or program upload requests coming from abnormal sources or at unexpected times/intervals.

The Comprehension step is used to collect information around observables and anomalies via the data sources identified in the previous step. AOOs should leverage various business organizations and areas of the OT environment for information for identifying and analyzing anomalies related to the Program Upload technique. This will allow them to more quickly identify triggering events using the Program Upload technique, enabling the decision making in the Decision step.

AOOs use the Decision step to determine a course of action based on an event that has occurred. With the information gathered, the AOO will be able to determine whether an anomalous program upload is indicative of an adversary’s presence in the network (thus initiating incident response procedures) or if there is no evidence of malicious activity associated with the anomalous program upload (thus initiating corrective maintenance procedures).

Additional assistance regarding general sensor placement and capability development is available through DOE; contact CyOTE.Program@hq.doe.gov for more information. AOOs can refer to the CyOTE methodology for more information on CyOTE's approach to identifying anomalies in an OT environment, which, when perceived, initiates investigation and analysis to comprehend the anomaly.

APPENDIX A: DATASETS TO SUPPORT TRIGGERING EVENT ANALYSIS FOR T845: PROGRAM UPLOAD

Table 5: Datasets to Assist with Analyzing Triggering Events

Dataset	Example Tools	Who Can Assist	Relevance
Netflow and Packet Data	● Wireshark/Tshark ● Commercial Passive Network Monitoring Tools (Claroty, Dragos, Nozomi, SilentDefense) ● Zeek ● NetworkMiner ● Snort ● Suricata ● Security Onion	● Network Security Team ● IT or OT System Admins	Netflow and packet data assists with identification of systems communicating and possibly detailed communication details
Device & System Logs	● SysInternals SysMon ● SysInternals PsLogList ● EvtxToElk ● Python-evtx ● OSQuery	● Network Security Team ● IT or OT System Admins	Device & system log data assists with identification of systems communicating and possibly detailed communication details
Device & System Configuration Files and Change History	SysInternals Suite	● Network Security Team ● IT or OT System Admins	Device & system configuration files and change history assists with identification of systems communicating and possibly detailed communication details
Account administration data like permission settings, account logs, onboarding information	SysInternals Suite	● Network Security Team ● IT or OT System Admins	Permission settings, account logs and onboarding information for accounts can assist with discovering other associated behaviors with the trigger event in question
Device or System Maintenance Documentation/Logs	SysInternals Suite	● Network Security Team ● IT or OT System Admins	Device or system maintenance document and logs assists with identification of systems communicating and possibly detailed communication details

Dataset	Example Tools	Who Can Assist	Relevance
Physical access logs and security monitoring data like CCTV output	Application Specific	Physical Security Team	Physical security logs and CCTV adds another factor of validation to assist with validation of true source
System engineering documents like network layouts and other schematics or diagrams	Diagram Specific	● Network Security Team ● IT or OT System Admins ● OEMs/Third-Party Vendors	Environment documentation assists with identification of other logging sources or impacted systems
Lists of software, apps, hardware, devices, or other relevant systems and their respective manufacturers	Asset Inventory Tools (Claroty, Dragos, Nozomi, SilentDefense)	● Network Security Team ● IT or OT System Admins	Software and hardware lists assist with identification of other impacted systems as well as other potential log resources to validate a trigger event
Any other data relevant to the investigation	Various	● Network Security Team ● IT or OT System Admins ● OEMs/Third-Party Vendors	Other data sources might contain information specific to a given trigger event

Click for More Information	CyOTE Program || Fact Sheet || CyOTE.Program@hq.doe.gov
DOE Senior Technical Advisor	Edward Rhyne || Edward.Rhyne@hq.doe.gov || 202-586-3557