T889: MODIFY PROGRAM

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 Modify Program attack technique for the Persistence 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 *Modify Program (T889) Technique Detection Capability Sheet* for the Persistence tactic.[4]

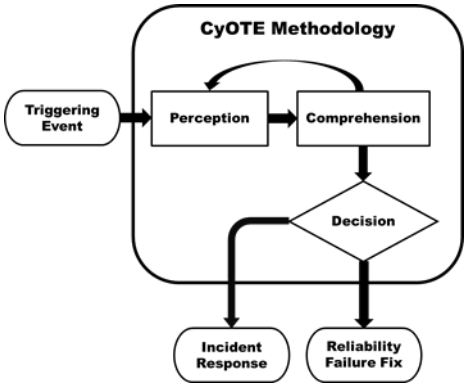

Figure 1: CyOTE Methodology Diagram

POTENTIAL ATTACK TARGETS

The Modify Program technique aims to change the run state of a controller. Adversaries may use the Modify Program technique to add or modify a program's instructions or logic on a controller to affect how it interacts with the physical process, allowing the adversary to potentially interact directly with the native application programming interface (API).[5] This change might occur via a program modification command within an industrial protocol or by using online edits or the *append* functionality. The functionality needed to execute this technique might take place via documented or undocumented protocols. Two examples of program modification attacks include Stuxnet's modification to Siemens' controller code and data blocks—which are downloaded to the PLC and adjusted based on the characteristics of the environment—and PLC-Blaster's ability to copy itself onto several program units on the target device.[6]

[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, Modify Program, 2021. Available online: https://collaborate.mitre.org/attackics/index.php/Technique/T0889.
[3] © 2021 The MITRE Corporation. This work is reproduced and distributed with the permission of The MITRE Corporation.
[4] CESER, Modify Program (T889) Technique Detection Capability Sheet, CyOTE Program. Department of Energy, FY21.
[5] MITRE, Modify Program, 2021. Available online: https://collaborate.mitre.org/attackics/index.php/Technique/T0889.
[6] Ibid.

PERCEPTION: IDENTIFYING ANOMALIES

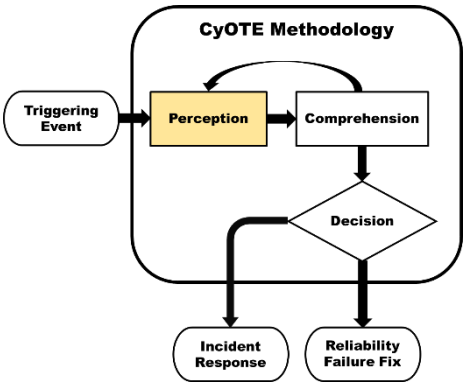

Figure 2: CyOTE Methodology – Perception Step

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[7] – 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.[8]

EXAMPLE OBSERVABLES AND ANOMALIES OF THE MODIFY PROGRAM 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 Modify Program technique.

Table 1: Notional Events

Observables	Anomalies	Data Sources
• An unexpected change in device operation state or behaviors • Increase in protocol function codes associated with control logic changes • An increase in unexpected device changes in application logs	Protocol messages executing modifications at abnormal frequencies or times of day resulting in a change of operational state	• Raw Network Data (Captured) • Raw Network Data (Live) • Application Logs

[7] Micah 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.
[8] 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
Analysis might identify an inconsistency between the physical or electronic characteristics of a device and the data within a historian or other diagnostic application	• Inconsistencies between alarm data and actual physical process diagnostics • Inconsistencies between the event recorder and physical process diagnostics • Unscheduled updates or changes made to control logic	• Application Logs • Raw Network Data (Captured) • Raw Network Data (Live)
For devices that count the number of program modifications on a device, an operator or security analyst might observe a change in the program counter	An unexpected change in the program increment counter on a device	• Application Logs • Raw Network Data (Captured) • Raw Network Data (Live)
• Analysis might look at device and application logs to identify changes in firmware and programs. Research might include log reviews focused on the update event itself or changes to metadata resulting from the firmware or program change. • Network data might also contain metadata changes.	An unexpected firmware or program update	• Application Logs • Raw Network Data (Captured) • Raw Network Data (Live)

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

Asset owners and operators aiming to detect potential capabilities to monitor for use of the Modify Program 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.[9]

1. Identify what devices and protocols to monitor for Modify Program
    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

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

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 for identified devices
    a. This may include servers, switches, security appliances, and logging locations (hosts)
        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[10, 11]
        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.

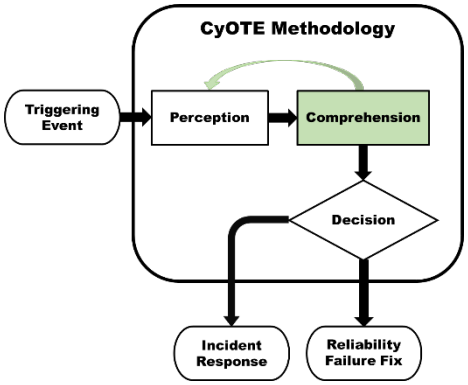

Figure 3: CyOTE Methodology - Comprehension Step

IDENTIFY AND COLLECT AVAILABLE INFORMATION RELATING TO PROGRAM MODIFICATIONS

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

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

information on an anomaly. Depending on the size and resources of your organization, some of these resources might not be applicable.

Table 2: Business Organizations That Support Information Collection for Modify Program

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 MODIFICATIONS

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 download message
- Program payload
- Payload size (e.g., bytes)

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

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 programs on devices of interest to alert on
   a. Document modification-initiated anomalies based on host
      i. Identify the existing traffic origination points
      ii. Include the frequency and type of modification(s)
   b. Identify and match high-risk program type(s) or magnitude for the physical process
      i. E.g., raising processing limits beyond established parameters or shutdown
      ii. Determine if the alert is valid or invalid based on analysis of the message parameters and source
2. Identify programs coming from new or abnormal hosts
   a. Analyze host lists for modifications to programs issues to end device(s)
   b. Conduct a comparative analysis to identify new connections and alerts versus old ones
   c. Determine whether messages are executing program modifications at an abnormal frequency
      i. E.g., frequency, order, type, messaging timing
      ii. Track messages and perform statistical and/or procedural tests
3. Establish triggers
   a. Incorporate the analysis findings provided previously and implement to refine alert parameters to focus on useful information and minimize the number of non-useful alerts
      i. E.g., new or abnormal messages, high-risk program modification messages, out-of-bound readings without alarms

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 Modifications

What To Report	Whom To Report To	Recommended Timeframe	Desired Outcome
Unexpected or unexplained change to industrial device or endpoint	• OT system administrators • OT engineers • Network security team	1 hour	Awareness of program change and further details to assist in perception and comprehension
An increase in program modification–related commands observed on the network or in host logs	• OT system administrators • OT engineers • Network security team	24 hours	• Awareness of program change and further details to assist in perception and comprehension • You should identify the cause for the increase in program modification traffic.

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 triggering condition(s). The order for which technical analysis should occur, or whether it is necessary for the comprehension stage, depends on the situation. Still, typically it will inform many of the context-building questions outlined in the following section.

Program modification might occur due to a local or remote change. The methods for how program modifications occur also vary by device vendor and configuration. Technical analysis should first attempt to understand the environment-specific conditions to gain proper understanding and situational awareness. Technical analysis should then focus on testing various hypotheses related to possible attack paths based on knowledge of the environment.

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:

- Are both remote and local program changes possible with the given device model and configuration?
- What log sources monitor program changes?
- What data retention limitations constrain the log sources?
- What is the standard local procedure for conducting program modifications? Was this event conducted within established guidelines, or does it appear to be initiated outside traditional practices?

Once the scope of possible hosts associated with the potential root cause is identified and data is collected, the analysis should focus on proving the original hypothesis developed through the original trigger event. Proving or disproving the trigger event hypothesis will support the decision-making process by validating initial perceptions and reducing initial cognitive biases.

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.

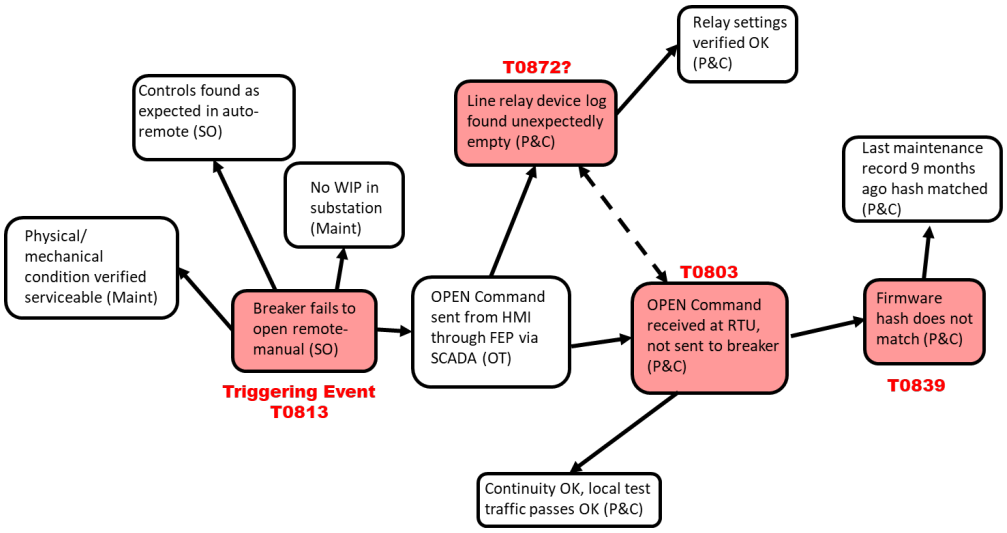

Figure 4: Example CyOTE Observables Link Diagram

INVESTIGATE POTENTIALLY RELATED ANOMALIES TO MODIFIED PROGRAMS

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.

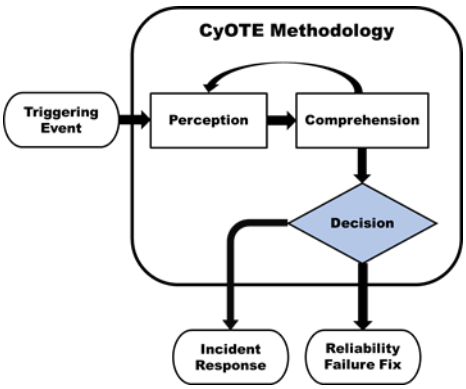

Figure 5: CyOTE Methodology - Decision Step

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 MODIFICATIONS

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
Firmware Verification	MITRE D3FEND: D3-FV[12]	Firmware verification validates that the current firmware matches the expected version. An attacker might change the firmware version also to modify the program. • Analysis should identify if a program modification occurred in conjunction with a firmware change. • Firmware verification provides one control to analyze potential low-level system changes.
Protocol Metadata Anomaly Detection	MITRE D3FEND: D3-PMAD[13]	Function code analysis or analysis of known ports and traffic volumes associated with program modifications provide a network-centric detection opportunity for program modifications. • Identify the various program modification paths supported in your environment. Program modification path analysis will require you to identify what protocols your particular system supports and has enabled.
User Data Transfer Analysis	MITRE D3FEND: D3-UDTA[14]	
Software Update	MITRE D3FEND: D3-SU[15]	Software updates might have an impact on running programs. Software to focus on includes human machine interfaces (HMI), engineering workstations (EWS), and other programs where the industrial application might contain software update functionality. • Tracking software updates assist with context-building during the comprehension stage. • Determine if changes to a running program occurred at the same time as a software update.
File Content Rules	MITRE D3FEND: D3-FCR[16]	File content rules and file hashing validate the integrity of a program or other logic going to an industrial device or endpoint.
File Hashing	MITRE D3FEND: D3-FH[17]	

[12] MITRE, Firmware Verification, 2021. Available online: https://d3fend.mitre.org/technique/d3f:FirmwareVerification.
[13] MITRE, Protocol Metadata Anomaly Detection, 2021. Available online: https://d3fend.mitre.org/technique/d3f:ProtocolMetadataAnomalyDetection.
[14] MITRE, User Data Transfer Analysis, 2021. Available online: https://d3fend.mitre.org/technique/d3f:UserDataTransferAnalysis.
[15] MITRE, Software Update, 2021. Available online: https://d3fend.mitre.org/technique/d3f:SoftwareUpdate.
[16] MITRE, File Content Rules, 2021. Available online: https://d3fend.mitre.org/technique/d3f:FileContentRules.
[17] MITRE, File Hashing, 2021. Available online: https://d3fend.mitre.org/technique/d3f:FileHashing.

Control	Matrix	Relevance
Protocol Metadata Anomaly Detection	MITRE D3FEND: D3-PMAD[18]	• Leverage known file hashes to track known programs on endpoints, share drives, and any other hash visibility points in the environment. • Hashes provide one source of validation and can also pair with other file content rules that look at the contents of a particular file. You might also examine file metadata to look for changes to a file.
File Carving	MITRE D3FEND: D3-FC[19]	Carving program files over network protocols enables network-centric detection of program changes. File carving adds additional context to supplement the function code analysis approach. • To validate a program with passive network analysis, you will need to be able to extract the program as it transits over the network. • File carving of programs from industrial protocols allows network-based visibility beyond program metadata analysis.

TESTING, DEPLOYING, AND REFINING ALERTING PARAMETERS FOR PROGRAM MODIFICATIONS

The parameters and established triggers 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)
        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
        iv. Use sandbox environment to test the functions
    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

[18] MITRE, Protocol Metadata Anomaly Detection, 2021. Available online: https://d3fend.mitre.org/technique/d3f:ProtocolMetadataAnomalyDetection.
[19] MITRE, File Carving, 2021. Available online: https://d3fend.mitre.org/technique/d3f:FileCarving.

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 Modify Program 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 Modify Program technique is being used as part of a malicious cyber-attack.

Through the Perception step, the AOO identifies observables and anomalies indicative of the Modify Program 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 Modify Program technique came to be. Protocol messages executing modifications at abnormal frequencies, inconsistencies between alarm data and actual diagnostics, and unscheduled updates to control logic are all potential observables that could indicate the use of the Modify Program technique. Anomalies tied to these observables could be unexpected changes in device operation state or behavior, an increase in unexpected device changes in application logs, or an inconsistency between the physical or electronic characteristics of a device and the data within a historian.

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 Modify Program technique. This will allow them to more quickly identify triggering events using the Modify Program 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 modification 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 modification (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 T889: MODIFY PROGRAM

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
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

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