T849: MASQUERADING

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 Masquerading attack technique for the Evasion tactic as defined by the MITRE ATT&CK® for Industrial Control Systems (ICS) framework.[2,3] This allows 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 *Masquerading* (T849) *Technique Detection Capability Sheet* for the Evasion tactic.[4]

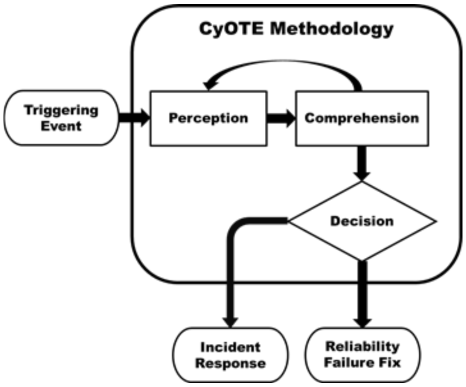

Figure 1: CyOTE Methodology Diagram

POTENTIAL ATTACK TARGETS

As defined by the MITRE ATT&CK® for ICS framework, adversaries may use the Masquerading technique to disguise malicious applications or executables as harmless files to avoid suspicion. The adversary may rename the files to imitate legitimate programs/program files, manipulate file metadata, and/or fool users into thinking that the malicious file is of a different type.

Devices that can be impacted by the Masquerading technique may include human-machine interfaces (HMI), engineering workstations, and any other devices that rely on executable code and/or files for programing and operation. Vulnerabilities may exist in software that can be used to disable or circumvent security features. Adversaries may use the Evasion tactic to exploit these software vulnerabilities to take advantage of a programming error in a program or service, or within the operating system software or kernel itself, to evade detection.[5] Various targets exist on all major operating systems such as dynamic-link library (DLL) files, installers, and commonly used application icons. These are frequent targets because of their ease of access and their ability to convince unaware users that they are in fact legitimate files or application shortcuts.

[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, Masquerading, 2021. Available from: https:/collaborate.mitre.org/attackics/index.php/Technique/T0849.
[3] © 2021 The MITRE Corporation. This work is reproduced and distributed with the permission of The MITRE Corporation.
[4] CESER, Masquerading (T849) Technique Detection Capability Sheet, CyOTE Program. Department of Energy, FY21.
[5] MITRE, Evasion, 2021. Available from: https://collaborate.mitre.org/attackics/index.php/Evasion.

PERCEPTION: IDENTIFYING ANOMALIES

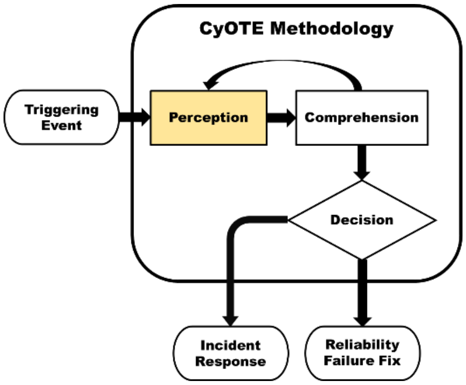

Figure 2: CyOTE Methodology – Perception Step

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." This nomenclature follows the North American Electric Reliability Corporation (NERC) nomenclature, which was adapted from Dr. Mica Endsley's model of situation awareness[6] – these terms 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; 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 MASQUERADING 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 Masquerading technique.

Table 1: Notional Events

Observables	Anomalies	Data Sources
Modified file extensions observed in network traffic, on a host, or in network security device alerts	• Executable files with unexpected file extensions • Communications where Multipurpose Internet Mail Extensions (MIME) type does not match	• Live or captured network traffic • Intrusion Detection System/Intrusion Prevention System • Manual

[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
	• Python library files (.pyd extension) have had their file extensions renamed	
Imitation of legitimate filename	• Disguised binary file "update.exe" • Suspicious or unexpected injection of compiled industrial program or industrial control system application binary	• Live or captured network traffic • Intrusion Detection System/Intrusion Prevention System • File signature pattern matching (YARA rules)
• Changes to expected command line arguments or process hierarchy associated with known application • Unusual or unexpected network communications from the renamed binary	• Past industrial attacks have used common sounding names (e.g., 101.dll, 104.dll, 61850.dll, and OPC.exe during CRASHOVERRIDE/Industroyer campaign) for malware. These malware samples, however, use very different command line parameters that might be visible in memory or in event logs. • The import/export table for malicious binaries might also contain suspicious or unusual parameters • Malware that masquerades with common sounding names might also have different parent and child processes associated with execution artifacts	• Files extracted from network traffic • Files analyzed on host • File signature pattern matching (YARA rules) • Process information from memory • Command line output from process enumeration tools (ps) and network connection enumeration tools that show process ownership information (netstat -anop on Windows and Linux)

STEPS FOR IDENTIFYING HIGH-CONSEQUENCE SYSTEMS, PATHWAYS, AND POTENTIAL ANOMALOUS ACTIVITY IN OT ENVIRONMENTS

Asset owners and operators aiming to develop potential capabilities to monitor for use of the Masquerading technique should consider a phased approach to development of the monitoring capability, to include continuous testing and evaluation throughout capability's life cycle. To complement this, it is highly encouraged to use the following steps to map out existing operational technology (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 that any defensive measures introduced do not negatively impact, adversely affect, or introduce vulnerabilities into the existing OT environment. As guidelines, the following best practices are recommended at a minimum:

1. Identify and compile a list of assets that are capable of performing or being targeted by Masquerading
    a. Identify the hardware and software configuration on assets to assist with identification of data sources available to support analysis. This may include logging functionality, enabled industrial and/or IT protocols, and polling frequencies, among others.
    b. Identify protocols in the environment, e.g., Ethernet/IP, S7Comm, Profibus, SERCOS III, Modbus, Distributed Network Protocol 3 (DNP3), host access protocol (HAP). Ensure identification includes both open-source and vendor-proprietary protocols.
2. Identify networked devices to be monitored for process state changes, e.g., programmable logic controllers (PLC), intelligent electronic devices (IED)
3. Identify data, logs, and log types needed to support identification of Masquerading from these key devices, including field devices
    a. Identify tap points to observe device network traffic
    b. Identify log stores on endpoints that contain important data relevant to the technique
    c. Include servers, networking switches, security appliances, and logging devices (hosts)
    d. Include logs that can be manually connected or sent to central log collection data stores
    e. Identify log retention timelines for each data source. Some devices might have rolling logs, so it is necessary to understand the capacity limit for when log sources roll over and how frequently that limit is reached in your environment. This might impact central log collection data stores and/or raw network data collection sources.
4. Identify business processes that support identification of Masquerading
    a. Identify opportunities where plant personnel or other network and device users would identify signs of technique occurrence
    b. Identify operational data stores that might assist with confirmation of technique identification
        i. Help desk tickets related to technique
        ii. Plant maintenance tickets related to technique
        iii. Unusual behaviors or actions that plant personnel or a network user would observe related to the technique

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

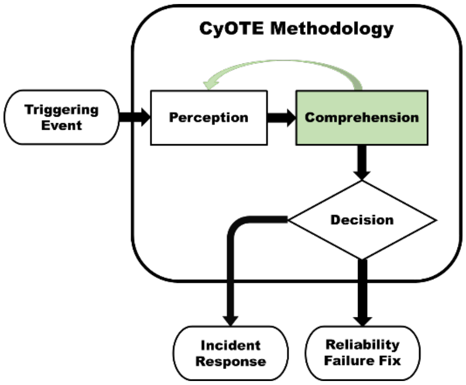

Figure 2: CyOTE Methodology - Comprehension Step

used for a sufficient investigation, comprehension for the purposes of CyOTE is an organizational ability, not an individual one.

IDENTIFY AND COLLECT AVAILABLE INFORMATION RELATING TO MASQUERADING

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.

Table 2: Business Organizations That Support Information Collection for Masquerading

Organization	Capacity
Cybersecurity Roles	Includes those responsible for the confidentiality, integrity, and availability of the organization's digital assets. These individuals provide a threat-informed perspective and bring experience and capabilities to analyze situations and data for cybersecurity issues.
IT Roles	Includes those responsible for the ownership, support, and administration of an organization's information technology assets.
OT Cybersecurity Roles	Includes those responsible for the support, administration, confidentiality, integrity, and availability of an organization's operational technology assets.
Original Equipment Manufacturers (OEM)	Includes those who produce and support the hardware and software present within the industrial environment. OEMs may be under support contracts but may provide technical documentation and expert advice on expected device behavior.
Third-Party Support Vendors	Cybersecurity, threat intel, and other solutions providers that provide subject matter expertise. These individuals may provide insight into anomalies surrounding 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 PARSING AND EXTRACTING INFORMATION FROM OT DATA FOR ANALYSIS OF MASQUERADING

The information on high-consequence systems, pathways, and potential anomalies identified 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.

Data collected here can be used for further analysis, outlined in the next section of this recipe. It is unlikely that any single AOO will have all of the data sources mentioned, this is a compilation of all possible data sources that would contain valuable data to identify the technique. Each individual AOO should collect what is available in their environment.

Suggested elements to collect include:

- File metadata
    - Timestamps
    - File types/extensions—common file types that adversaries attempt to imitate include .pdf, .jpg, .png, .txt, .exe, .bat, .gif, and document/folder icons, among others
- File Hash Data
- Patch/Update schedule
    - Identify files modified outside of scheduled patch/update windows
- Device identifier(s) (will vary based on environment)
    - IP addresses
    - Source and destination ports
    - MAC addresses
    - Operation(s) to be watched

Suggested logs to collect include:

- Controller logs
    - Controller log files such as syslog, when available
- Operating system logs
    - Syslogs
    - Windows Management Instrumentation (WMI) logs
    - Windows event logs
- Network management software
- Security information and event management (SIEM)/security orchestration, automation, and response (SOAR)
- Network security monitoring

STEPS FOR ANALYZING ANOMALIES FOR MASQUERADING

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 upon. Any message revealing two or more parameters is given greater precedence for analysis and correlation with other observables to identify potential indications of the Masquerading technique.

Monitoring for indications of the Evasion tactic includes monitoring sensitive files for integrity and any unscheduled/unauthorized file modifications. Additionally, monitoring device logs for file modifications and state-change entries, or changes occurring at irregular times and/or through compromised accounts can alert on potential Evasion tactic activity.

1. Identify traffic coming from new or abnormal hosts
    a. E.g., an engineering laptop being used outside of business hours or operational maintenance windows
    b. Devices communicating that haven't previously communicated
2. Analyze system for evidence of file and/or file metadata modification
    a. Timestamps being modified
    b. Files being renamed to imitate other legitimate files
    c. Unauthorized users modifying files or performing unusual edits
    d. Changes to file hashes
3. Identify other tactics, techniques, and procedures (TTP) to use for correlation of potential incidents
    a. Invalid code signatures: Adversaries may attempt to deceive users by mimicking features of valid code signatures pulled from a legitimate, signed, program
    b. Right-to-left override: Adversaries may append a right-to-left override character to a filename to trick users into executing a malicious file that appears benign
    c. Renaming of system utilities: Adversaries may rename system utilities to attempt to evade security mechanisms that may otherwise be used to detect their activities. Additionally, legitimate files/programs may be copied and/or moved to non-standard directories, which may result in the ability to evade detection due to the utility being executed from a non-standard path.
    d. Masquerading tasks or services: Adversaries may attempt to disguise malicious tasks/services by naming the malicious tasks/services similarly to the legitimate ones
    e. Files that match legitimate names or locations: Adversaries may name/place malicious files or resources in seemingly legitimate locations to avoid detection. In certain environments, they may also create resources in namespaces that match the naming convention of the container.
    f. Space after filename: Adversaries may append a space after a file extension thereby changing how the file is processed by the system; this can be used to trick a user into clicking on a seemingly benign file and unintentionally executing something malicious.

g. Double file extension: Adversaries may attempt to conceal malicious files from users by appending a secondary extension to a seemingly benign file. The second extension may be hidden from view from the user without more thorough examination and are often seen in spearphishing email attachments.

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 time frame 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 Masquerading

What To Report	Whom To Report To	Recommended Timeframe	Desired Outcome
Phishing email	• Cybersecurity Department • IT Department	Immediate, 1 hour	• Email to be quarantined & sender blocked; should trigger analysis of destination computer in case of malicious file download. Any attachments should be analyzed in a secure environment.
File extension renaming	• Cybersecurity Department • IT Department • Engineering Department • Systems Operations Department	Immediate	• Should trigger incident response plan - assume complete network breach

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 change to a device's operating mode 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). Technical analysis for masquerading should focus on the circumstances of the observed file/metadata modifications and any associated human or system actions that might have initiated the file/metadata changes. A change to file attributes such as name, extension, or metadata might be manually initiated by a user or programmatically through software/malware. Because of the potential for file renaming to be a normal, benign workplace occurrence, it is important to investigate the origin of the changes to gain better understanding of who or what caused it in the first place.

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:

- Who has the appropriate access and permissions to initiate changes to the affected files?
- What changes were observed, and do they correlate with known TTPs?
- Where on the network were devices with modified files identified and were there any other affected devices adjacent to the original affected device?

- Was any unusual network traffic detected/recorded around the suspected time of the incident?

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

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.

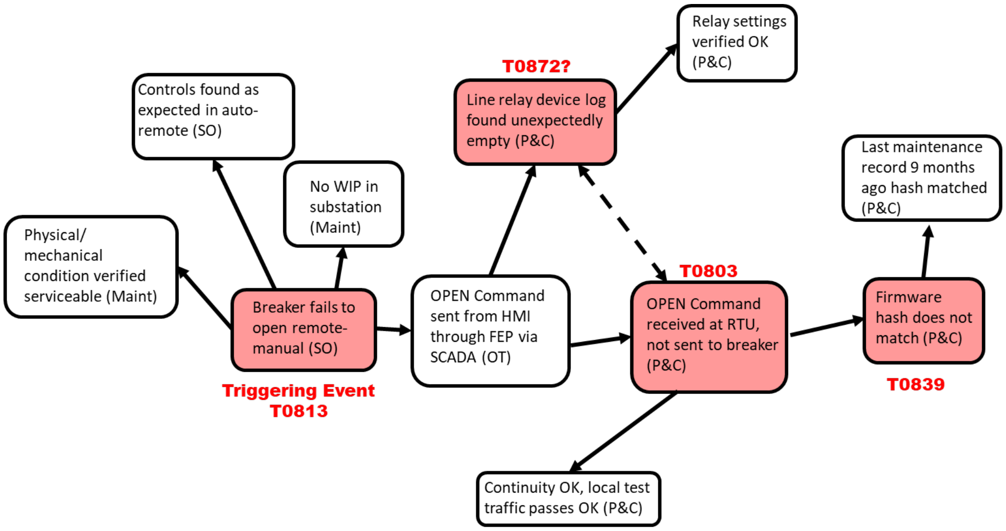
Relay settings verified OK (P&C)
T0872?
Line relay device log found unexpectedly empty (P&C)
Controls found as expected in auto-remote (SO)
Last maintenance record 9 months ago hash matched (P&C)
No WIP in substation (Maint)
Physical/ mechanical condition verified serviceable (Maint)
T0803
Breaker fails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

Figure 4: Example CyOTE Observables Link Diagram

A worm diagram showing the use of the Masquerading technique in the 2017 Triton attack on the Petro Rabigh refinery complex in Rabigh, Saudi Arabia is shown in Figure 5.[8]

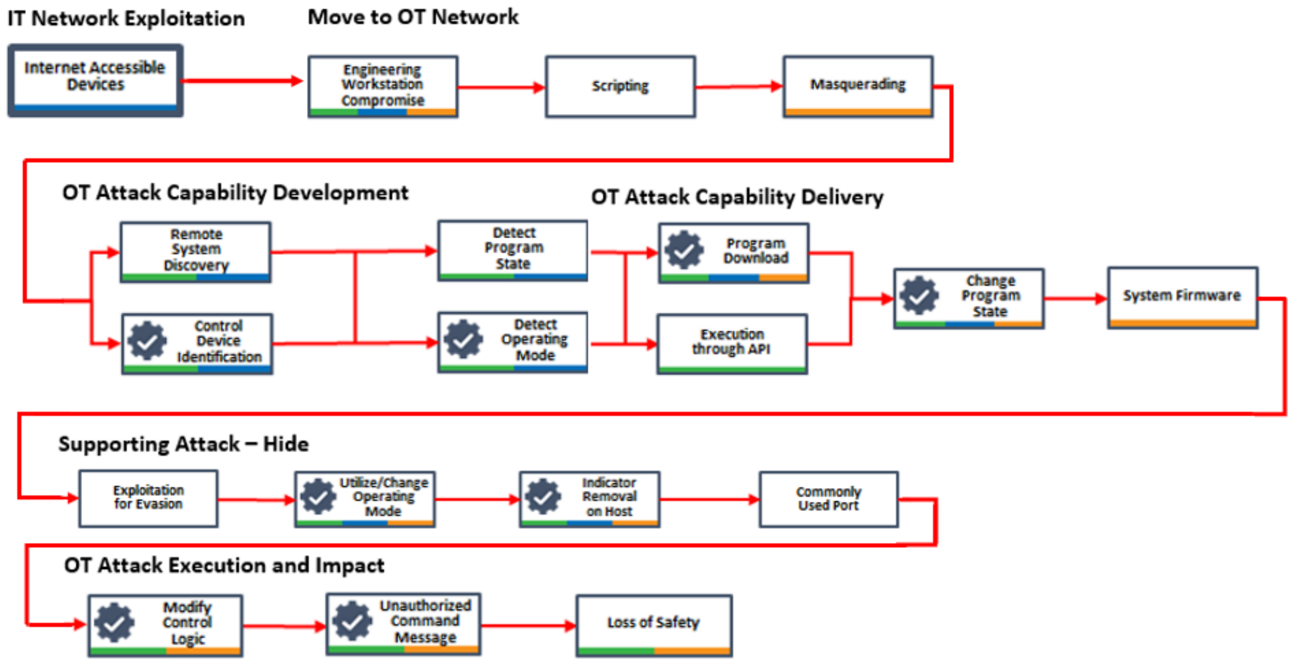

Figure 5: CyOTE Observables Link Diagram in Triton Case Study

INVESTIGATE POTENTIALLY RELATED ANOMALIES TO MASQUERADING

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.

[8] CyOTE Case Study: Triton in Petro Rabigh, https://inl.gov/wp-content/uploads/2021/09/Triton-CyOTE-Case-Study.pdf
A legend for this diagram is included in the CyOTE Case Study: Trion in Petro Rabigh

DECISION: REMEDIATING THE CAUSE AND EFFECTS OF A TRIGGERING EVENT

The last step of the CyOTE methodology is the Decision step (Figure 6). 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.

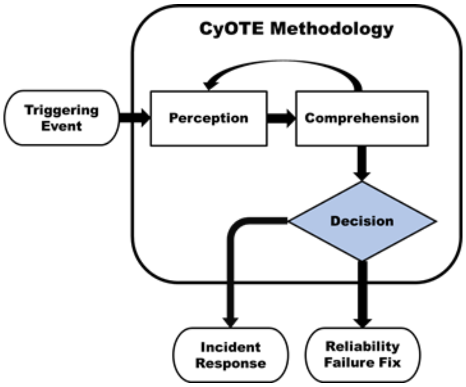

Figure 6: CyOTE Methodology - Decision Step

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 MASQUERADING

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[9]	Used in conjunction, network segmentation, network traffic community deviation, and protocol metadata anomaly detection 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 but can also include host-based application and operating system logs associated with the network traffic. ● Leverage knowledge of the network environment to understand what devices are running particular processes that may be susceptible to disruption via file modification/masquerading ● Proper network segmentation provides functional boundaries where monitor and block actions can be implemented to prevent unauthorized changes to files from unauthorized subnets ● Understanding of what normal file metadata looks like in your environment is critical for analysis of abnormal file manipulation
Network Traffic Community Deviation	MITRE D3FEND™: D3-NTCD[10]	
Code Signing	MITRE ATT&CK for ICS: M0945[11]	Code signing provides a mechanism to maintain binary and application integrity by using/verifying digital signatures which helps to prevent untrusted code from executing.
Execution Prevention	MITRE ATT&CK for ICS: M0938[12]	Limiting code execution on a system through application control, and/or script blocking provides a way to lock down systems and protect against malicious code execution.

[9] MITRE, Network Segmentation, 2021. Available from: https://collaborate.mitre.org/attackics/index.php/Mitigation/M0930.
[10] MITRE, Network Traffic Community Deviation, 2021. Available from: https://d3fend.mitre.org/technique/d3f:NetworkTrafficCommunityDeviation.
[11] MITRE, M0945: Code Signing, 2021. Available from: https://collaborate.mitre.org/attackics/index.php/Mitigation/M0945.
[12] MITRE, M0938: Execution Prevention, 2021. Available from: https://collaborate.mitre.org/attackics/index.php/Mitigation/M0938.

Control	Matrix	Relevance
Restrict File and Directory Permissions	MITRE ATT&CK for ICS: M0922[13]	Restricting access to files/directories by enforcing file system access controls makes it more difficult for would-be attackers to access/move/modify files in protected locations.
Strong Password Policy	MITRE D3FEND: D3-SPP[14]	Strong passwords provide a line of defense against both insider threat and external threats. Strong passwords should be used on all endpoints in order to protect misuse of the applications capable of conducting operating mode changes. Without a strong password policy, an attacker might use weak credentials to compromise an account and change the operating mode of a device remotely.
Audit	MITRE ATT&CK for ICS: M0947[15]	Account auditing validates the permissions of existing accounts and review of access logs. Proper account auditing provides a host-based baseline monitoring opportunity. • Identify and limit what roles are required for users on endpoints that regularly make operating mode changes • Ensure proper logging exists for attempts to act outside of a user's expected roles
Privileged Account Management	MITRE ATT&CK for ICS: M0926[16]	
Local Account Monitoring	MITRE D3FEND: D3-LAM[17]	
Access Management	• MITRE ATT&CK for ICS: M0801[18] • NIST 800-53: SI-7(8)[19] • NIST 800-53: AC-220 • NIST-800-82: 6.2.3[21] • NIST-800-82: 6.2.17[22]	

TESTING, DEPLOYING, AND REFINING ALERTING PARAMETERS FOR MASQUERADING

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

[13] MITRE, M0922: Restrict File and Directory Permissions, 2021. Available from: https://collaborate.mitre.org/attackics/index.php/Mitigation/M0922.

[14] MITRE, Strong Password Policy, 2021. Available online: https://d3fend.mitre.org/technique/d3f:StrongPasswordPolicy.

[15] MITRE, Audit, 2021. Available online: https://collaborate.mitre.org/attackics/index.php/Mitigation/M0947.

[16] MITRE, Privileged Account Management, 2021. Available online: https://collaborate.mitre.org/attackics/index.php/Mitigation/M0926.

[17] MITRE, Local Account Monitoring, 2021. Available online: https://d3fend.mitre.org/technique/d3f:LocalAccountMonitoring.

[18] MITRE, Access Management, 2021. Available online: https://collaborate.mitre.org/attackics/index.php/Mitigation/M0801.

[19] NIST, NIST Special Publication 800-53, Revision 5, "Security and Privacy Controls for Information Systems and Organizations,"2020, available online: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf.

[20] Ibid.

[21] NIST, NIST Special Publication 800-82, Revision 2, "Guide to Industrial Control Systems (ICS) Security," 2015, available online: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf.

[22] Ibid.

are realized during this process, the products of which are sent to output destination(s) in selected format(s).

1. Identify logging data required to perform alerting
    a. Network traffic captures
    b. Endpoint process execution logs
2. Where feasible, identify opportunities to ingest logging data into an alerting platform (e.g., SIEM, endpoint detection and response [EDR]/network detection and response [NDR], Datalake). Ensure collection balances collection of data that supports remote service alerting and analysis with operational network stability limitations and constraints. Some industrial OEMs have preferred products to accomplish this designed for different industrial control systems.
    a. Network tapping and data
    b. Endpoint native or third-party log forwarders to SIEM
3. Where feasible, Implement logging and data collection equipment and configurations. If not feasible, identify opportunities to supplement logging and data collection with processes and technologies that accomplish similar outcomes.
    a. SIEM
    b. EDR
    c. NDR
    d. Windows Event Forwarding (WEF)/Syslog Forwarding
4. Create alert(s) to monitor for activity
5. Implement protection/prevention capability/configurations
    a. Reference the Control Matrix in Table 4 of this Recipe for different protection and prevention opportunities
    b. Ensure that selected protection and prevention controls fit the parameters of your environment and don’t degrade or interfere with operations

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

Through the Perception step, the AOO identifies observables and anomalies indicative of the Masquerading 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 Masquerading technique came to be. Unscheduled file modifications that and/or the discovery of modified files are potential observables that could indicate the use of the Masquerading technique. Anomalies tied to these observables could be executable files with unexpected file extensions, python library files with renamed file extensions, or suspicious or unexpected injection of compiled industrial program or industrial control system application binary.

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 Masquerading technique. This will allow them to more quickly identify triggering events using the Masquerading 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 file 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 file (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 T849: MASQUERADING

Table 5: Datasets to Assist with Analyzing Triggering Events

Dataset	Example Tools	Who Can Assist	Relevance
File Hash Data	● Certutil ● Sha256sum ● Shasum ● Powershell ● Hash Generator	● Network Security Team ● IT or OT System Admins	File hash data will provide insight into whether or not files have been changed, assisting with timeline generation
Event Logs	● LogParser ● Event Log Explorer ● LOGalyze	● Network Security Team ● IT or OT System Admins	Event logs assist with root cause identification and timeline generation
Device & System Configuration Files and Change History	● Sysinternals Suite ● Engineering Workstation or HMI Software	● Network Security Team ● IT or OT System Admins	Device & system configuration files and change history assist with root cause identification and timeline generation

Click for More Information	CyOTE Program | | Fact Sheet | | CyOTE.Program@hq.doe.gov