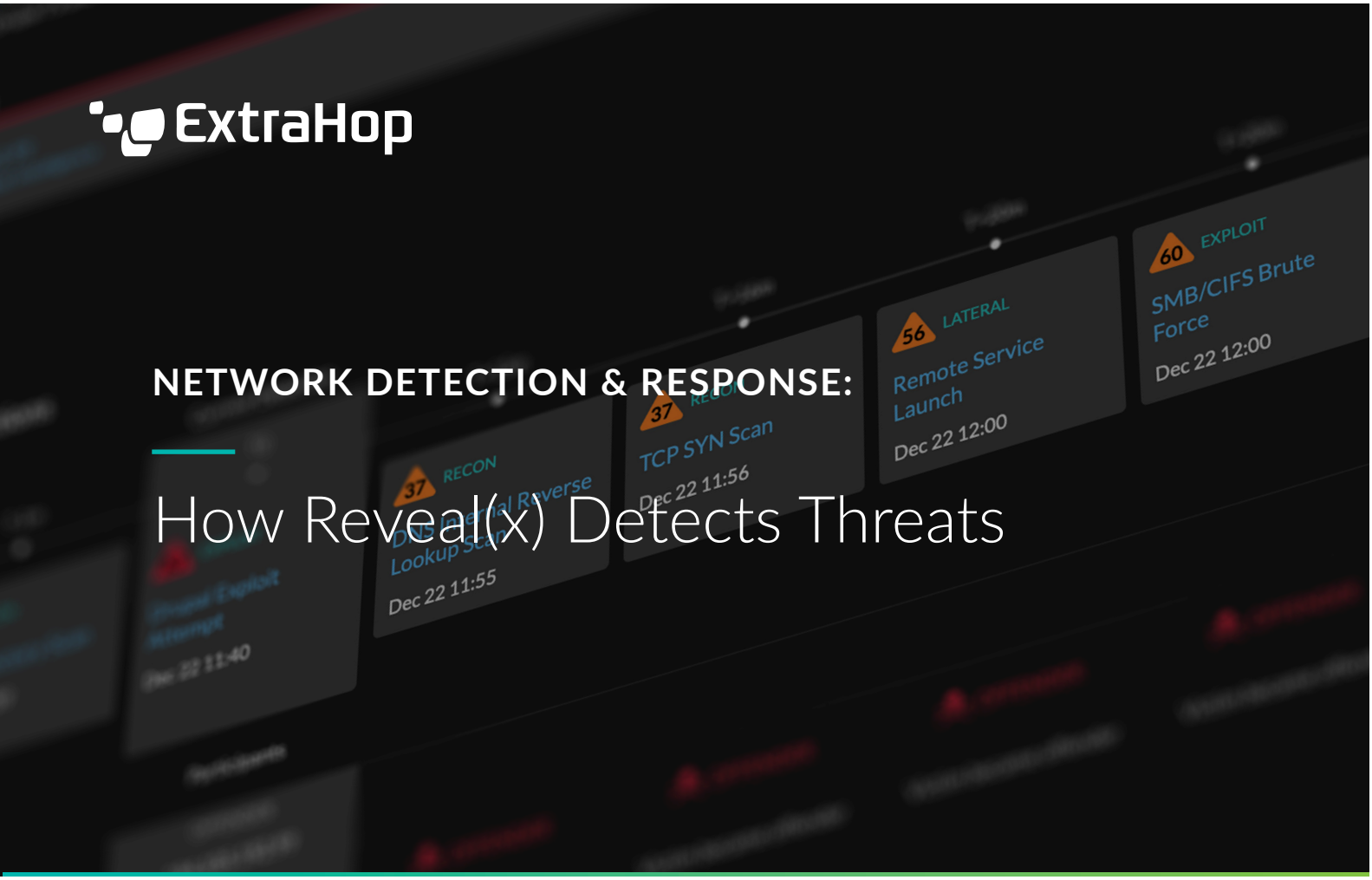
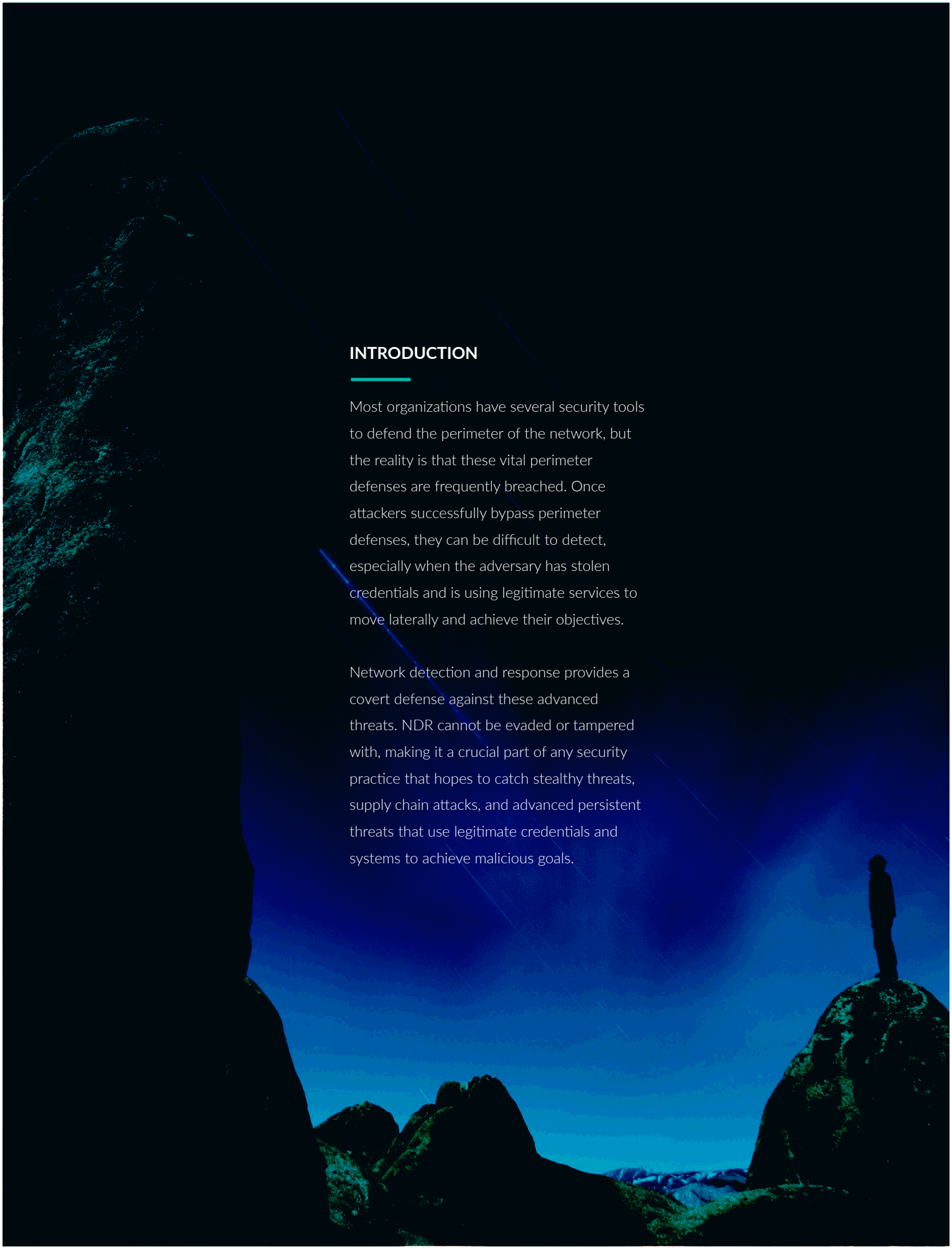


EXECUTIVE SUMMARY

The purpose of this paper is to provide a clear explanation of how ExtraHop Reveal(x) network detection and response (NDR) provides post-compromise detection capabilities, as well as enhanced perimeter security, with more accurate detection capabilities than traditional intrusion detection systems (IDS).

This paper will explore the technical details of how Reveal(x) NDR enables security teams to resolve threats 84% faster, using a full-spectrum detection approach that combines real-time detection of the latest CVEs and continuous behavioral machine learning to catch stealthy, post-compromise attacker tactics, techniques, and procedures.

TABLE OF CONTENTS



INTRODUCTION

Most organizations have several security tools to defend the perimeter of the network, but the reality is that these vital perimeter defenses are frequently breached. Once attackers successfully bypass perimeter defenses, they can be difficult to detect, especially when the adversary has stolen credentials and is using legitimate services to move laterally and achieve their objectives.

Network detection and response provides a covert defense against these advanced threats. NDR cannot be evaded or tampered with, making it a crucial part of any security practice that hopes to catch stealthy threats, supply chain attacks, and advanced persistent threats that use legitimate credentials and systems to achieve malicious goals.

HOW REVEAL(X) DETECTS THREATS

> “ExtraHop Reveal(x) decreased time to threat detection by 50 percent, and time to threat resolution by 84 percent.
>
> **FORRESTER TOTAL ECONOMIC IMPACT REPORT**

The network is an ideal point for detecting these post-compromise attack activities for several reasons:

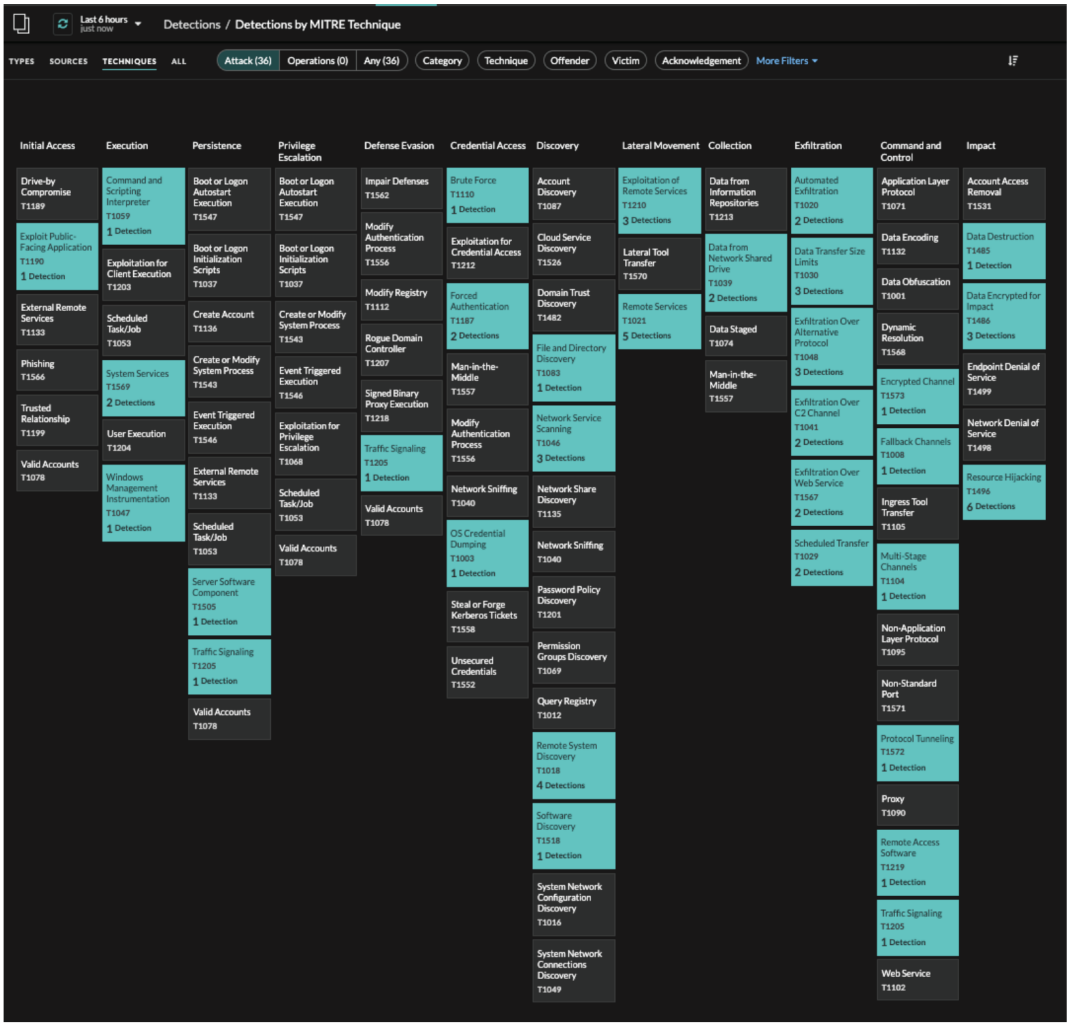
- **The network offers covertly observed ground truth.** While log data and endpoint data offer value for threat detection, they can be turned off, evaded, or modified. Passively observed network traffic is not subject to this tampering. Attackers have no way to be sure whether their network activity is being observed.
- **Attacker behaviors and techniques on the network are not subject to the same high variability and polymorphism that plague the endpoint.** There are millions of ways for attackers to get remote code execution capability on a host and steal data, but there is only a handful of ways to exfiltrate that data across the network. Watching for those network behaviors is much more likely to catch the attacker. Network signals can be used to detect many attacker tactics, techniques, and procedures across every stage of the MITRE ATT&CK Framework.
- **The network offers greater breadth and depth of visibility than other options.** Any device that communicates across the network can be discovered and monitored immediately when it first appears. Details such as users, software, operating systems, and more can be discerned from the device's network communications.

Reveal(x) provides both enhanced perimeter detections, more accurate than traditional intrusion detection, as well as a vital additional layer of defense in the case of an intrusion by detecting activity throughout the lifecycle of an attack. This includes detecting subtle post-compromise activity such as misuse of Windows remote procedure calls and abnormal behavior from low-privileged devices or users. This behavior-based detection approach is able to effectively detect attacks with a much lower false-positive rate than legacy, signature-based intrusion detection systems, which are known for being too noisy to provide much value, but which are still hanging on in many enterprise environments. Reveal(x) detects anything an IDS can detect, and much more, with greater context and confidence, providing coverage for many attacker tactics, techniques, and procedures across every category of the MITRE ATT&CK Framework, which is directly integrated into the product, as illustrated.

> One of our worst nightmares is that out-of-band network tap that really is capturing all the data, understanding anomalous behavior that's going on, and someone's paying attention to it. You've gotta know your network. Understand your network, because we're going to.

ROB JOYCE, FORMER HEAD OF NSA TAILORED ACCESS OPERATIONS

Joyce is speaking from the point of view of one of the world's most sophisticated hacking organizations.



The MITRE ATT&CK Framework is integrated into the Reveal(x) NDR interface.

This white paper explains how Reveal(x) detects threats at all stages of the attack lifecycle. Here, we'll explain the different types of detectors that we offer for reconnaissance, exploitation, lateral movement, command and control, network privilege escalation, and data exfiltration, among other attacker tactics, techniques, and procedures.

- Reconnaissance
- Initial Exploitation
- Establish Persistence
- Install Tools
- Move Laterally
- Collect Exfil and Exploit

SPECTRUM OF DETECTIONS

Reveal(x) employs numerous methods, including rule-based detection, machine-learning behavioral analysis, peer group analysis, and deep learning to detect the full spectrum of attack activity. Instead of relying on a single method, this combination of techniques provides more holistic coverage of attacker tactics, techniques, and procedures.

	Hygiene Activity violates security policy, deviates from secure practices, or otherwise indicates risk. For example, ports, protocols, and services that are vulnerable or non-compliant.	**Known Attacks** Includes IP addresses, domains, file names, payload strings, or protocol behavior that have been observed in past attacks.	**Unknown Attacks** Attacks that do not have a previously known identifier, but exhibit anomalous behavior that can be linked to a part of the attack lifecycle.
Built-in rule-based	Rule-based detections identify security hygiene issues by comparing observed behaviors on the network against security best practices. This uncovers risks such as the use of the vulnerable SMBv1 protocol or password information sent cleartext, and more.	Similar to intrusion detection systems (IDS) rules, Reveal(x) uses complex rules logic to identify known attacks by specific attributes. Most CVEs that rely on the network can be identified in this manner, along with techniques used in lateral movement. Developed by ExtraHop threat researchers, these rules-based detections are delivered and updated through the cloud.	Not applicable.
Custom rule-based	Organizations can also create custom rule-based detections that identify policy violations, such as cleartext data movement between network segments housing PII or health data, that may be unique per environment. Reveal(x) includes a scripting capability for real-time parsing of enterprise protocols.	Customers can build detectors to detect threats and mitigate risks unique to their own environment or business, assuring total coverage.	Custom rule-based detections can be used as “tripwires” protecting critical assets such as important file servers and databases. Organizations can create custom rules-based detections that fire when a device begins connecting outside a prescribed IP range or across network segments.
Machine Learning	Not applicable.	Reveal(x) uses machine learning to model behaviors of entities on the network and contextually identify behaviors that resemble known attack techniques.	Reveal(x) employs a variety of machine learning techniques to detect malicious activity based on anomalous behavior, including deviations from peer group activity, privilege escalation, and more.

Rule-based Detections

Reveal(x) uses rule-based detection, similar to what a traditional intrusion detection system might use, except that Reveal(x) recognizes behavioral patterns, not MD5 hashes or other easily evaded signature methods. Take the TrickBot malware as an example. An IDS signature-based detection approach might search only for specific, hard-coded strings that appear in the modules of the program, such as "</dinj>", which appears in the dllinject module of some variants of Trickbot. This is an easy string to change. The attacker can rename that string to anything they want, and evade detection. A behavior-based approach to detecting TrickBot might rely on observing characteristics of the SSL connection initiated by the malware, such as using a very recently created, Self-Signed certificate with null Cert Issuer and Cert Subject. These behaviors are much harder for the attacker to change while still achieving their goals with TrickBot. This behavioral rule-based approach can identify known tactics and techniques, such as CVE exploit attempts and hacking tools, in ways that are much harder for attackers to evade than a simple string-matching signature approach. The logic behind the detections is developed by ExtraHop threat researchers and delivered to sensors through cloud updates. Built-in detectors are continuously updated via the cloud to assure all customers have the most recent detectors as quickly as possible when new CVE detections become available.

Because Reveal(x) decrypts traffic and inspects the transaction payload, the threat research team can access strong indicators of an attack (such as the post_render form field of an HTTP request) that would be obscured if the traffic could not be decrypted or the analysis did not include transaction payload details.

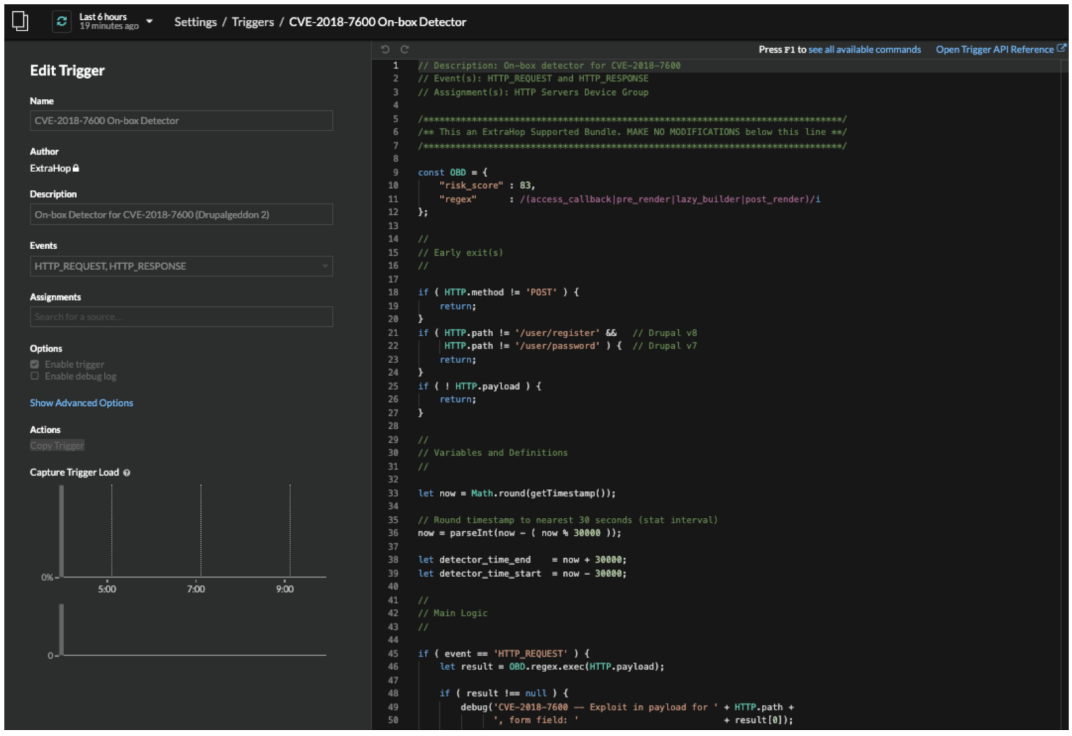
*Reveal(x) examines details such as the **post_render** field of HTTP transactions to reliably detect vulnerability exploit attempts.*

> Threat detection today needs local context a lot more than people realize.
>
> **ANTON CHUVAKIN, FORMER GARTNER ANALYST**

Custom Rule-based Detections

Even with hundreds of out-of-the-box detectors, no network detection and response (NDR) solution can meet every unique organizational requirement. To accommodate each organization's needs, Reveal(x) enables users to create custom detections with a sophisticated scripting engine. For example, organizations can create detections for policy violations specific to individual servers or server clusters that host critical applications or data. Custom detections can be mapped to the MITRE ATT&CK Framework as well so that they are displayed in the MITRE ATT&CK Matrix visualization within the product interface.

The Reveal(x) scripting engine provides access to complex logic applied to stateful, real-time analysis of network traffic—including access to 5,000+ built-in metrics, 50 event types, and direct access to the transaction payload for custom parsing. Users can write triggers in JavaScript to create their own custom detections to handle needs unique to their organization.



Machine Learning

The promise of machine learning for detection is that it can catch attack behaviors that evade detection by rule or signature-based methods, and provide higher accuracy for detection of known attack tactics, techniques, and procedures (TTPs).

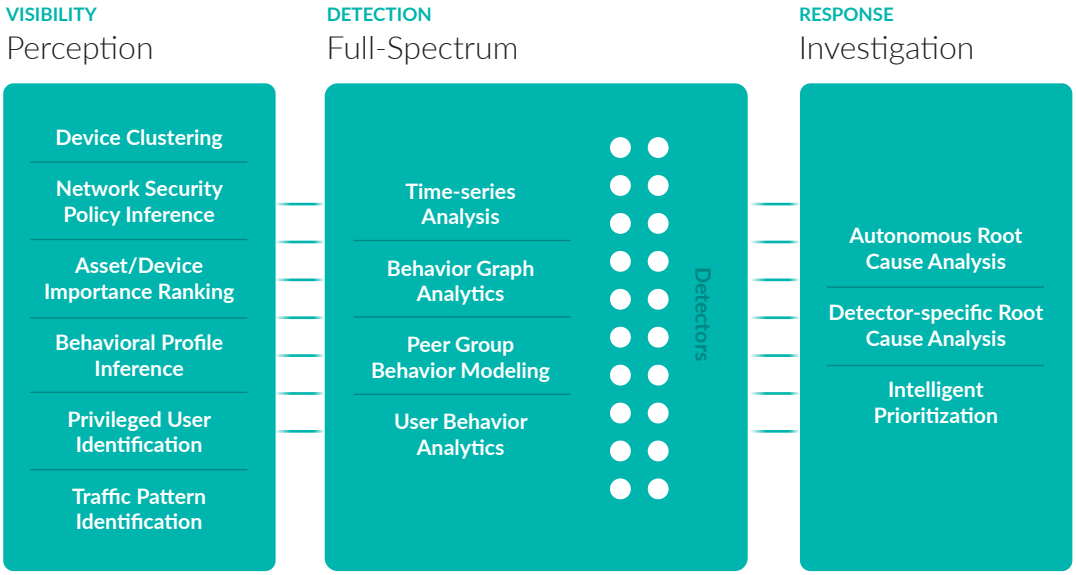
Increasingly, attackers are "living off the land," meaning that they are using approved services to mask their activity. When an attacker blends in by using legitimate credentials and approved services, how are traditional IDS tools going to detect them? That's where machine learning comes in. Machine learning detects attack activity inside an IT environment by modeling the behavior of each entity and contextually identifying behaviors that resemble known attack techniques. For example, PsExec is a common admin utility used by IT admins. PsExec is also used by attackers, but alerting on every instance of PsExec usage would be noisy, and not useful. Reveal(x) ML observes the historical behavior of devices on the network and identifies when an instance of PsExec usage is unexpected and malicious based on context clues, including device ownership, device role, organization, time of access, and more.

Reveal(x) leverages ML-powered analysis and detection delivered by three subsystems: Perception, Detection, and Investigation.

Reveal(x) uses ML to detect attacks and help analysts investigate and respond to attacks faster by automating information gathering and putting that information into context.

To achieve this, ExtraHop builds its ML using a multi-subsystem design, very similar to modern autonomous driving solutions, where a collection of sophisticated and patented ML subsystems—designed to extract insights, detect threats, and gather context—work in unison.

Reveal(x) leverages ML-powered analysis and detection delivered by three subsystems: Perception, Detection, and Investigation. Each subsystem contains multiple components responsible for distinct functions. ML components in the same subsystem and across different subsystems collaborate and exchange data and findings:



For example, Reveal(x) ML analyzes network traffic patterns to understand what the privileged assets are in your environment. This allows Reveal(x) to detect when a device performs abnormal privilege escalation. These ML capabilities can also be used to automatically discern how critical a given device is by watching its behavior and access patterns, dynamically updating the risk score of detections involving critical devices.

In addition, Reveal(x) can also tell when a device starts behaving differently than is normal for its peers, for example, when a conference room phone unit tries to initiate an SSH session, which no other conference room phone unit has ever done in your environment.

An important advantage machine learning has over IDS-style rules-based detection is the ability to learn what is normal. Whereas IDS detection is based on static indicators present in network traffic over a short time, Reveal(x) understands behavior over different time windows using predictive models, enabling the solution to more accurately detect activity such as data staging and exfiltration, and to answer the question, "Is it normal to see database traffic on this client?"

> "We don't have better algorithms than anyone else; we just have more data.
>
> **PETER NORVIG, DIRECTOR OF RESEARCH, GOOGLE**

Reveal(x) uses detection cards to provide analysts with one click comprehensive forensic level details. Additionally, detection cards map known attack types to the MITRE ATT&CK framework, giving analysts insights such as what stage of an attack they are dealing with.

For machine learning detections in Reveal(x) that use these predictive models, the detection card helps analysts understand why the detection fired by providing details about the behavior that triggered the detection. This helps to make Reveal(x)'s machine learning mechanisms less obscure and gives users greater confidence.

Doesn't Everyone Have Machine Learning Now?

No. While excessive use of the terms machine learning and artificial intelligence in marketing materials can make it seem like ML is ubiquitous, not every product uses these techniques. Even more importantly, not all ML is created equal. There's a broad spectrum of quality among ML approaches in cybersecurity.

In general, three capabilities distinguish ExtraHop's implementation of machine learning for detection versus other security products:

1. **ExtraHop uses more numerous and high-quality machine learning features:** The effectiveness of machine learning is heavily dependent on the machine learning features that feed into the algorithms. These features include network protocol behaviors and interactions such as database transaction methods, SQL queries, user behaviors, and thousands more. Reveal(x) extracts 5,000+ features from network traffic, making them available for machine learning. ExtraHop sensors are purpose-built to extract these features for our ML system at up to 100Gbps.

> Without the ability to see into encrypted traffic in the network environment, analysts are effectively "flying blind." Analysts can configure Reveal(x) to monitor encrypted traffic, including traffic protected by Perfect Forward Secrecy.
>
> **DAVE SHACKLEFORD, SANS PRODUCT REVIEW OF REVEAL(X)**

2. **ExtraHop leverages the speed and scalability of the cloud:** Reveal(x) relies on the scalable computing resources of the cloud to continuously train and execute hundreds of machine learning models. This cloud-scale machine learning is unique in the NDR market, and it enables Reveal(x) to deliver much more sophisticated and robust detections than competitors that perform machine learning locally on the sensor appliance, where compute resources are limited. In addition, hosting the service in the cloud means that ExtraHop can rapidly deploy updates.
3. **ExtraHop can decrypt more:** Reveal(x) will decrypt encrypted traffic up-to and including SSL/TLS 1.3-enrypted traffic so that it can access Layer 7 application details, such as error messages and other payload information, making this highly relevant information available for machine learning features. To facilitate the capability, Reveal(x) integrates with Domain Controllers to obtain all necessary encryption keys in real time, ensuring full visibility and protocol parsing for all Active Directory protocols, such as LDAPS, RDP, MSRPC, WMI, SMBv3, SIP-over-TLS, and many more. ExtraHop supports both NTLM and Kerberos encryption configurations for Microsoft Environments.

    a. There are cases where decryption is not possible, or customers will choose not to decrypt traffic. In order to provide the best threat detection capabilities possible, Reveal(x) also performs analysis of encrypted traffic for threat detection and response. Reveal(x) uses machine learning algorithms to pinpoint malicious patterns in encrypted traffic to help identify threats and improve incident response.

For more information about how Reveal(x) cloud-scale machine learning works, read our blog posts: *Tricks of the Trade: How Reveal(x) Uses Machine Learning* and *ExtraHop Cloud Scale ML: A Deep Dive*

Curated Data to Augment Detections

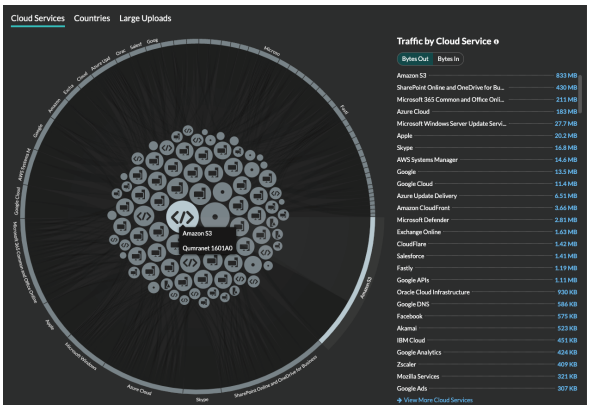
On top of the core dataset of network traffic, fully reassembled, decrypted, and analyzed in real time, Reveal(x) uses several additional data sources to augment the detection of attack activity. This augmentation data is delivered to Reveal(x) appliances automatically through cloud updates.

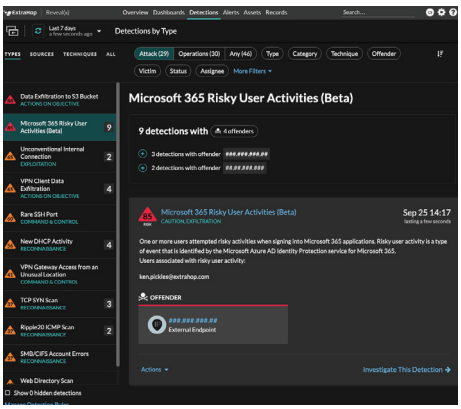
- **Threat intelligence** - Reveal(x) includes a curated threat intelligence feed, but users can also utilize their own STIX-formatted threat intelligence feeds. Matches are displayed throughout the Reveal(x) UI.
- **Ransomware definitions** - Like other malware, ransomware strains have indicators including domains, IP addresses, the file name of the ransom note, and unique file extensions given to the encrypted files.
- **Tor node updates** - Reveal(x) monitors IP addresses associated with Tor nodes, which are used to mask attackers' identities.
- **Hardware device fingerprinting** - Reveal(x) identifies specific device makes and models by using a variety of techniques, including analysis of DHCP traffic and usage of a half-dozen other protocols.
- **Cloud Service Fingerprinting** - By leveraging known IPs and hostnames, Reveal(x) is able to identify cloud services such as DropBox, Google Drive, AWS, Azure, and more.
- **Software and OS fingerprinting** - Almost all OS's and software packages that leverage the network have unique fingerprints that can be used to identify the OS and software versions, often with sufficient detail to specify major and minor software builds.

Monitoring SaaS Services

The Reveal(x) platform identifies and monitors traffic to a larger number of SaaS-based services, such as Amazon AWS & S3, Google Cloud, DropBox, Azure Drive, and many more. This monitoring allows customers to track all traffic going to and from SaaS-based services, aiding network administrators in identifying utilization patterns, and eliminating access to unauthorized services. This data is also used to detect malicious activities, such as data exfiltration to cloud storage. The data is also used to add context and enrich investigations, providing comprehensive visibility into behavior on the network.

In the case of Microsoft 365, Reveal(x) integrates directly via API to enable detection and investigation of risky Microsoft 365 user behaviors through the intuitive Reveal(x) interface.





Microsoft 365 Integration for Enhanced Threat Detection

Reveal(x) 360 is able to surface and display security detections from Microsoft 365, including risky user behavior and risky login detections. Reveal(x) provides rich network context around these detections to help analysts rapidly investigate and remediate threats. Many security analysts face the frustrating challenge of needing to look at multiple, isolated security tools to investigate events in the environment. This adds friction and slows down investigations, giving attackers time to expand their access. Integration between Reveal(x) 360 and Microsoft 365 brings together detections, investigative workflows, and forensics into a single interface.

Additionally, Reveal(x) offers monitoring and decryption of Microsoft Active Directory traffic, SMBv3, MSRPC, winRM, and other Microsoft protocols that are often targeted by attackers. With the addition of Microsoft 365 monitoring, Reveal(x) 360 offers security teams in hybrid enterprises an unprecedented level of visibility and investigative capability without having to pivot between tools. This integration detects and correlates risky user behaviors and logins, driven by observed behaviors, such as:

- **Impossible Travel:** When a user signs in from two geographically different locations.
- **Password Spraying:** A type of brute force attack
- **Suspicious Inbox Forwarding:** Identification of suspicious email forwarding rules
- **Known compromised credentials:** User credentials have been found in a published database of stolen materials
- **Communication with malicious IPs or domains:** A user connects from, or interacts with, a known malicious IP address or domain
- And many more

LOWERING THE EXPERTISE BARRIER FOR ANALYSTS

In an ideal world, every Security Operations Center (SOC) would be fully staffed by analysts with years of expertise and unfailing intuition. Of course, the reality is different. Most security organizations are understaffed and in the process of training up more junior analysts. ExtraHop empowers SOC analysts to confidently validate detections and understand the potential severity of an event within one or two clicks. Reveal(x) accomplishes this by automatically gathering contextual information unique to each detection and presenting it to analysts via interactive detection cards.

Risk score	Prioritizes the detection based on its severity.
Mitre ATT&CK Mapping	Highlights which MITRE ATT&CK tactic, technique, or procedure a detection relates to.
Time and duration	Shows how the detection relates to other events chronologically. If the event is ongoing, a special tag is appended to the detection indicating so.
One-click filters for detections with the same participants	If the Offender or Victim has been involved in other detections, the detection card will show how many and which types of detections they were. Additionally, the card provides a link to a list of detections with the same participants.
Ticket number, status, and assignee	Pulls service ticket information so that analysts do not have to search in the ticketing system.
Attack description	Descriptions of the attack including a brief summary of why the detection fired, how the attack works, and what the objective might be.
Offender and victim details	Device details for the Offender and Victim devices, including roles, device name, IP address, and links to activity maps.
Related assets	Lists affected users, devices, or files with links to the full metadata details. Knowing what data and devices are involved in this event helps analysts to ascertain its potential severity.
Metric Details	For detections based on predictive models, the protocol details (such as method and status code) are shown with a sparkline for the protocol activity, the baseline for the period, and peak value.
Investigation steps	Live Activity Map visualization of all traffic between devices. Automatically tailored metric or record searches showing the devices, files, usernames, or transactions associated with the detection. Automatically filtered peer-group behavior comparisons. One-click link to full packet capture. These searches and filters retrieve more detailed data that often helps analysts confirm a detection event.
Risk score factors	Calculation methodology involves the probability of the attack happening, the skill level required, and potential business impact.
Detailed attack background	An explanation of the attack, including how it works and possible criminal motivations, that is helpful for less experienced analysts or other team members who are not security experts. This information saves analysts time that would otherwise be spent looking online for answers.
Mitigation options	For each detection, a list of potential mitigation options is offered to help prevent or mitigate the risk of that particular attack.
Reference links	Links to authoritative third-party web pages such as the MITRE ATT&CK Framework, CVE database, OWASP, and more resources describing the tactic or technique. One of the most common activities for analysts investigating a detection is searching online for information, and these links eliminate that step and assure access to credible information.
Feedback mechanism	Analysts can mark detections as helpful or not, providing instant, actionable feedback to ExtraHop's threat research and data science teams.

Background data about this attack type, risk level,mitigation recommendations, and links to MITRE ATT&CK listings for tactics used in this detection.

Victim and Offender data, includingdevice role and IP address

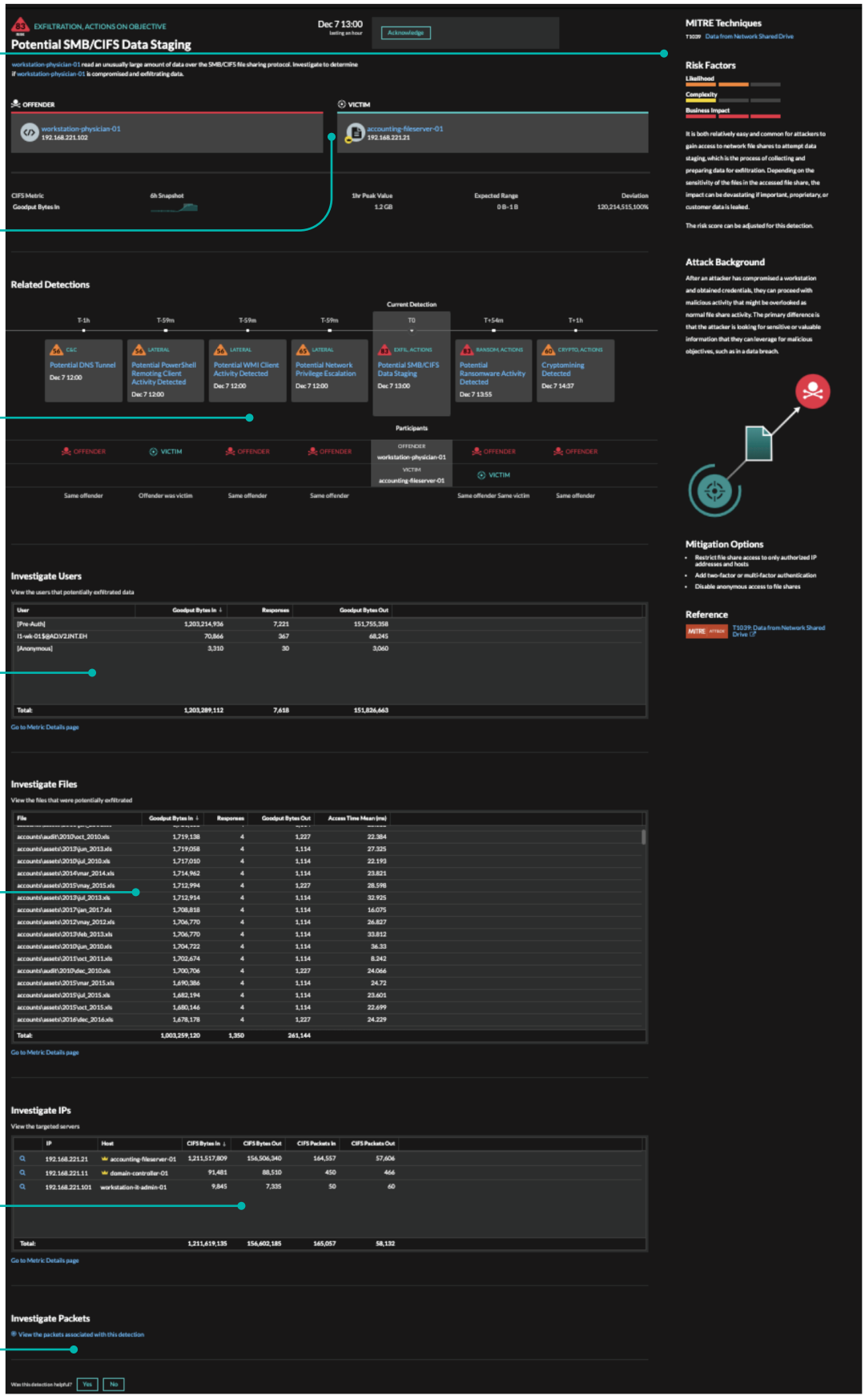
Related detections timeline showing what happened with the victim and offender before and after this detection.

Users involved in the detection

List of files that were accessed ormanipulated in the course of the behavior that triggered the detection.

List of IP addresses involved in the detection. Clickable for further investigation.

One click access to the relevant packets for forenisic investigation.



Easily Understand Device Contextual Information

Reveal(x) is also able to identify asset types and functions with a great degree of detail, including, gleaning device criticality based on behavior analysis, and gathering such information as:

- Which operating systems are in use on each device, and what role each device plays. For example, Reveal(x) can tell whether a device is a domain controller, web server, DNS server, IoT webcam, or conference room phone (among many other roles) simply by observing each device's behavior.
- Which users have logged in, or attempted to log in to each device?
- Which, if any, EDR agent is installed on the device?

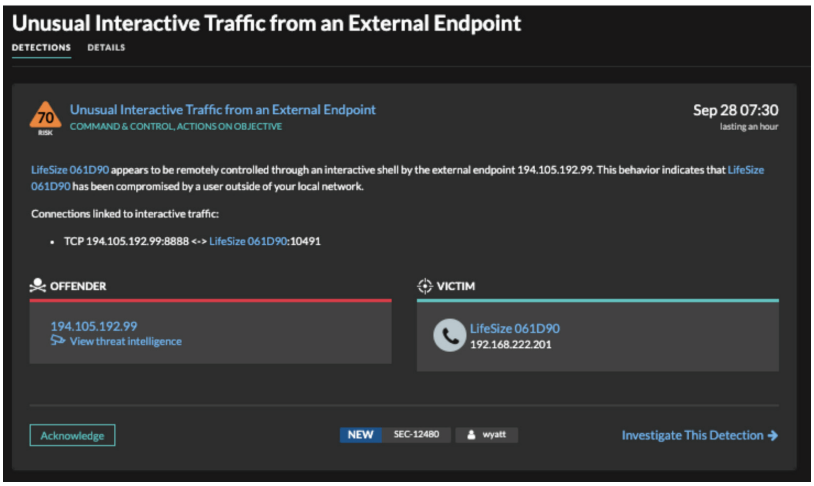
These capabilities enable Reveal(x) to compare behavior within and across peer groups of devices. If a single conference room phone starts to exhibit behavior that no other conference phone has exhibited, that will be flagged as suspicious. Comparing suspicious behaviors among peers, and understanding whether an attack is primarily targeting IoT devices or some other category or role group, helps analysts understand the scope of an attack and accelerate their response and mitigation activities.

Detection Validation Workflow Examples

UNUSUAL INTERACTIVE TRAFFIC FROM EXTERNAL ENDPOINT

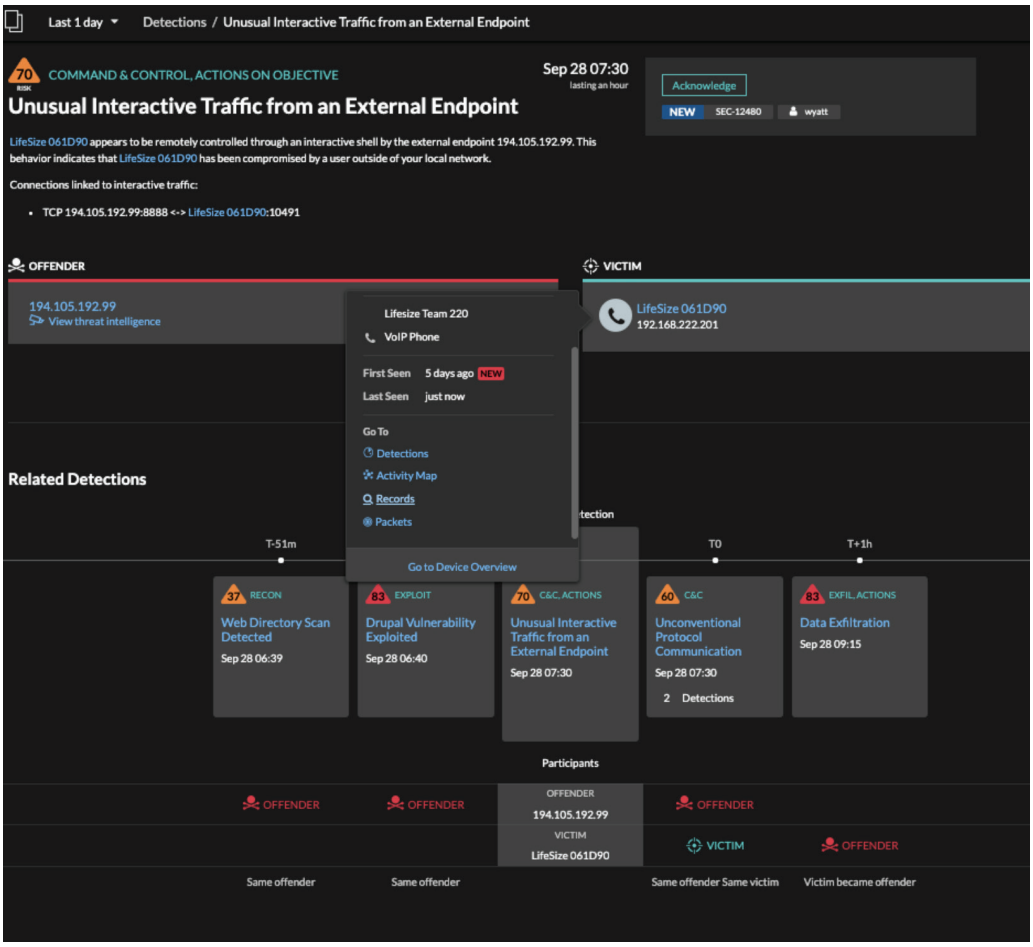
BEGIN WORKFLOW

Click "Investigate This Detection" for more details and context.



See related detections, links to MITRE ATT&CK and OWASP documentation, and threat intel details on the detection card.

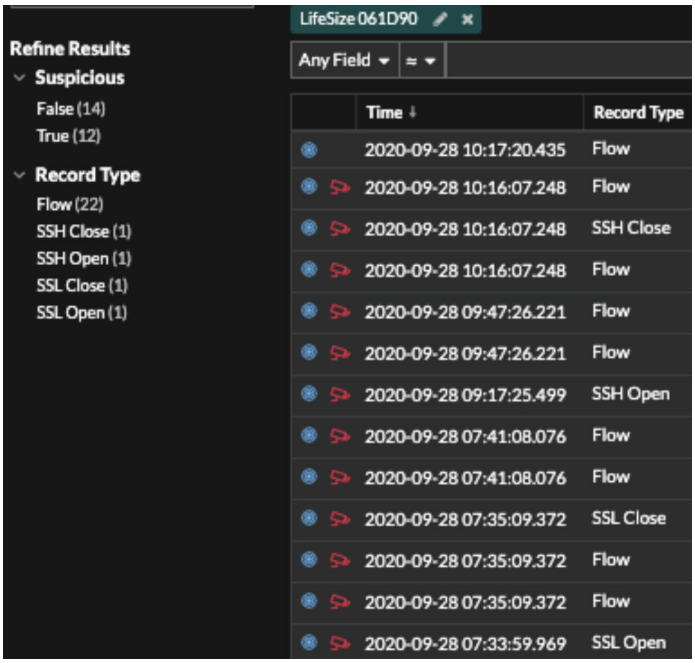
Hover over the Offender or Victim to access full transaction records or packets for the traffic that caused the detection.



The transaction records show multiple rapid SSH and SSL opens and closes from an IP address that matched our threat intelligence feed.

This validation took just two clicks from the initial detection.

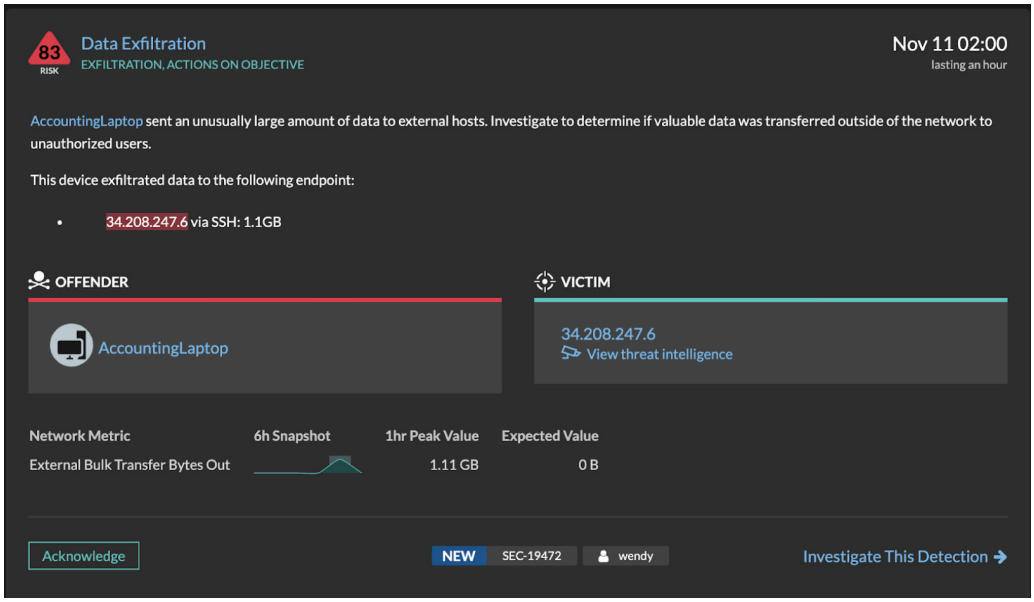
END WORKFLOW



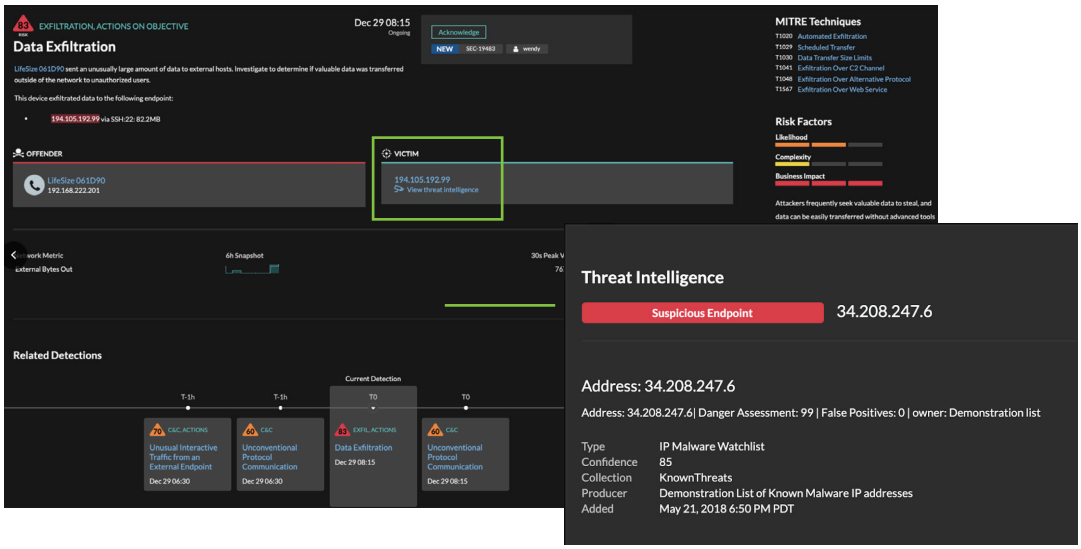
DATA EXFILTRATION

BEGIN WORKFLOW

Click "Investigate This Detection" for more details and context.

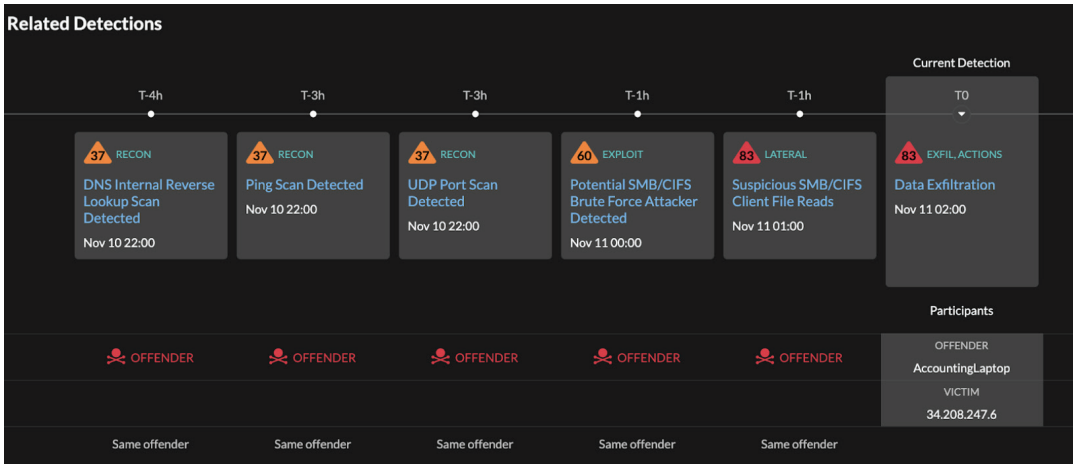


Click "View threat intelligence" for 3rd party threat feed associated data.

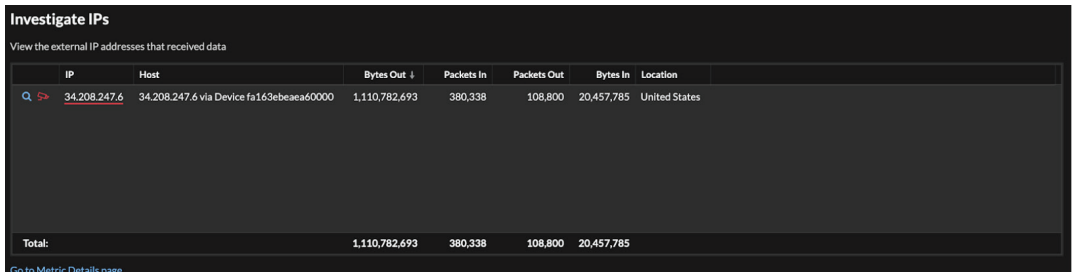


Scroll to see related detections, links to MITRE ATT&CK and OWASP documentation, and threat intel details on the detection card.

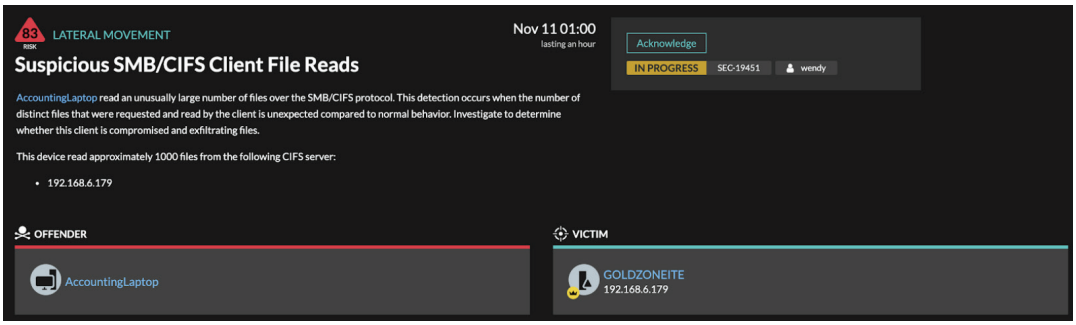
Note the Suspicious SMB/CIFS Client File Reads immediately prior to the Data Exfiltration Detection.



Review IP addresses and Hosts associated with the detection to understand what the traffic quantity looks like and where it was sent.

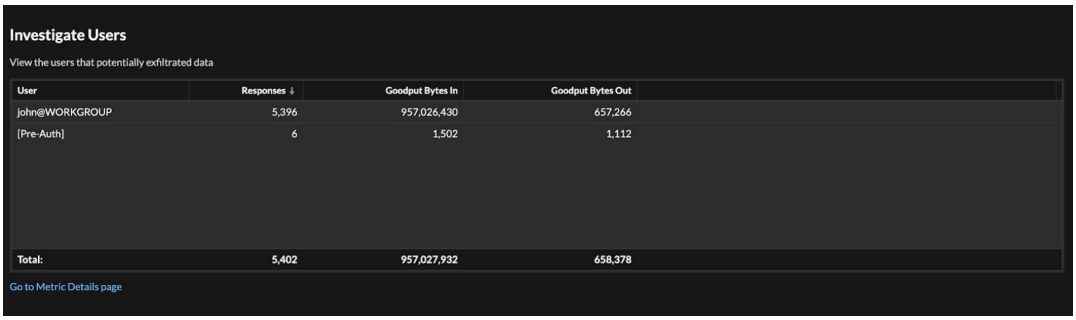


Click on "Suspicious SMB/CIFS Client File Reads" in the related detections timeline. This shows that the AccountingLaptop interacted



Based on the SMB/CIFS traffic Reveal(x) is able to provide the user that was used to authenticate to the server providing context around what accounts may be compromised.

Scrolling down to the Investigate Files section provides a detailed list of the files that were read from the GOLDZONEITE server.

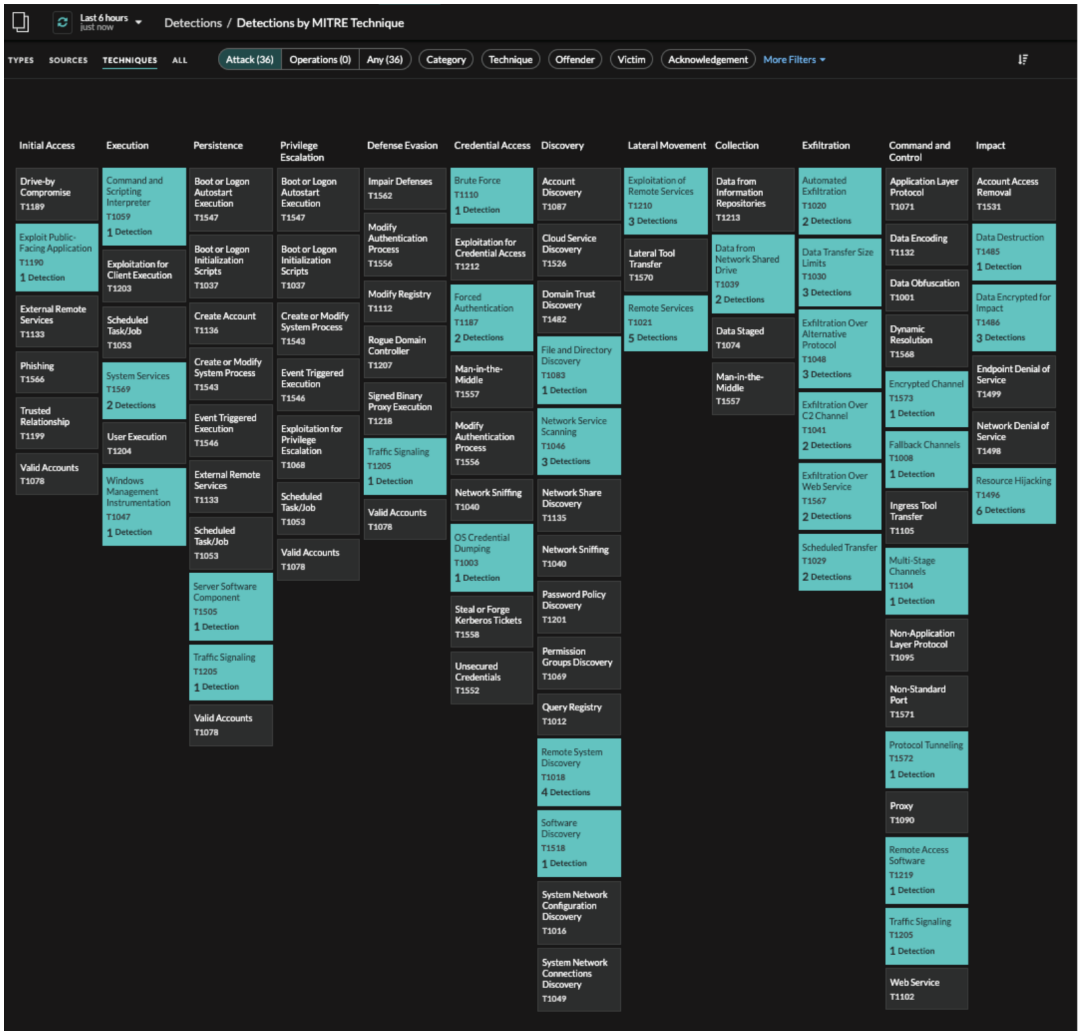


END WORKFLOW

MITRE ATT&CK Matrix is Built Right In

Part of reducing friction for analysts to investigate and validate alerts is to provide as much context as possible. In addition to the information included in every detection card, Reveal(x) places detections within the broader context of the MITRE ATT&CK Framework. Analysts can see which of the attack stages each detection falls into and which attack techniques have been detected along the attack chain to quickly understand attacker activities in the network during the selected time window. These detections are included with Reveal(x) and work right out of the box, but analysts may also use the custom detection builder to create detections and map them to the MITRE ATT&CK Framework.

The MITRE ATT&CK Matrix uses a more detailed evolution of the Lockheed Martin Attack Chain to identify which part of an attack each detected technique belongs to.



Two Perspectives On MITRE ATT&CK Coverage:

Technique Type and Category

Reveal(x) provides two primary ways to filter detections based on the MITRE ATT&CK framework Technique Type and Asset Category.

Filtering by Technique Type displayed all the ATT&CK techniques related to current detections along with a count of how many detections are related to the technique. Analysts are then able to specify exactly which attack technique they are interested in and view detections related to the specific technique.

Filtering by Category allows analysts to evaluate detections in terms of attack stage. Detections are filtered based on attack stage regardless of the type of asset the detection is related to. For example, Reveal(x) detections for data exfiltration can be mapped to IoT devices as well as Windows endpoints. These detections are valid for both managed and unmanaged devices.

CONCLUSION

Detecting threats is more challenging now than ever before. Attackers have adapted to evade legacy detection mechanisms such as hash signatures and string literal matching. They've learned to route around or disable activity logging and other data sources.

Network detection and response leveraging machine learning behavioral analysis provides the most confident detections, and is the most difficult for attackers to evade. Don't take it from us, though, just listen to Rob Joyce, former leader of the NSA's Tailored Access Operations division, who said: "One of our worst nightmares is that out-of-band network tap that really is capturing all the data, understanding the anomalous behavior that's going on, and someone's paying attention to it. You've gotta know your network. Understand your network, because [attackers] are going to."

Experience Reveal(x) for yourself with our free online demo

You made it to the end!

But really, this is just the beginning. To learn more about how Reveal(x) NDR can solve key security use cases for your organization, check out these blog posts or head to our free online demo (the only one of its kind in the NDR market).

The Tricks of Our Trade: How Reveal(x) Uses Machine Learning to Detect Threats

ExtraHop Cloud-Scale ML: A Deep Dive

Supervised vs. Unsupervised Machine Learning: Which is Better for Threat Detection, and Why?

Putting Machine Learning to Work for Enterprise IoT Security

APPENDIX A

Examples of Application-Layer Machine Learning Features

Most NDR products offer limited application-layer (Layer 7) analysis, which means that they do not have access to highly relevant machine learning features. Below are some examples of the application-layer features available to Reveal(x) users, along with examples of how they are used to detect malicious behavior.

Note: Reveal(x) parses many more application-layer protocols than those listed below. These examples illustrate the types of machine learning features that are only available after decryption and full-stream reassembly.

Database Application-Layer (Layer 7) Features: Methods, Errors, Requests, Responses, SQL Statements

Example of why it matters: Database methods and error messages are necessary for confident detection of brute-force attacks against the database server.

Kerberos Application-Layer (Layer 7) Details: User Principal Names, Service Principal Names, Request and Response Message Types, Error Types

Example of why it matters: Kerberos duplicate ticket errors are a key component in accurately detecting when an attacker has stolen a ticket and is replaying it. Kerberos requests asking for a service principal name (SPN) that does not exist are crucial to detecting username scans.

DNS Application-Layer (Layer 7) Details: Record Types, Response Codes, Host Queries

Example of why it matters: DNS is frequently misused by attackers to conduct reconnaissance (e.g. scanning) and command and control communications (e.g. tunneling). In these cases, it is necessary to have visibility into DNS host queries and DNS record types.

LDAP Application-Layer (Layer 7) Details: Errors, Requests, Responses, Plain Text Messages, SASL Messages, Bind Distinguished Names, Search Query

Example of why it matters: LDAP errors and account names are important features when detecting malicious authentication activity such as brute-force attempts.

SMB/CIFS Application-Layer (Layer 7) Details: Errors, File Paths, Username, Requests, Responses, Reads, Writes, Warnings

Example of why it matters: CIFS write activity and file names are extremely relevant features when detecting ransomware, and logon error messages are important when detecting brute-force attacks.

SSH Session-Layer (Layer 5) Details: Record Type, Version, Cipher Algorithm, MAC Algorithm, Compression Algorithm, KEX Algorithm, Client Implementation, Server Implementation

Example of why it matters: SSH algorithm details are necessary for accurate detections of suspicious tunneling activity.

SSL/TLS Session-Layer (Layer 5) Details: Versions, Alerts, Content-Type, JA3 Hash, Certificate Subject, Certificate Expiration Dates, Domains (SNI), Cipher Suites, Record Sizes

Example of why it matters: SSL/TLS certificate issuer details are extremely relevant details for accurately detecting malicious encrypted traffic.

APPENDIX B

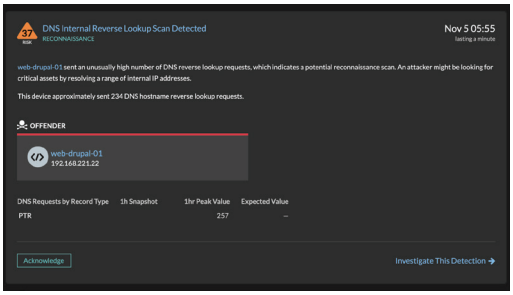
Examples of Detections in Reveal(x)

Attackers can easily modify the tools they use to avoid signature-based detection, but it's nearly impossible to avoid engaging in certain network behaviors required to achieve their goals. Reveal(x) uses a spectrum of techniques, including ML-powered behavioral analysis, to detect malicious activity across every category of the MITRE ATT&CK Framework. The examples below demonstrate the breadth of coverage provided by Reveal(x).

Note: These are example categories of detections, not a comprehensive list.

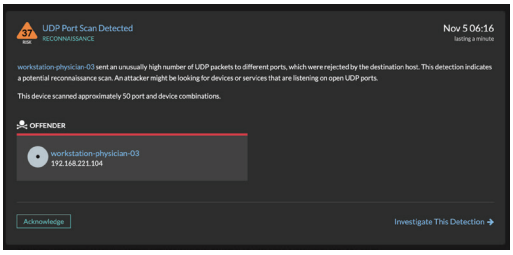
Reconnaissance

Before they ever send a phishing email or attempt a web exploit, sophisticated attackers need to gather information about their target. However, this reconnaissance activity does not stop once they gain access to a system. Once inside a network, attackers continue to reconnoiter the environment to orient themselves and determine what steps they need to take next to achieve their overall objective. Attackers often perform scans of the network to discover devices, services, file servers, and directories. This activity can look similar to legitimate vulnerability scanning. Reveal(x) detects a wide variety of scanning activity, however, many companies leverage vulnerability scanners internally which is why Reveal(x) applies heuristics to identify known vulnerability scanners so that analysts can more easily determine if the scan is approved or not. Detections for scanning activity from known vulnerability scanners can be suppressed in Reveal(x).



DNS Internal Reverse Lookup Scan Detected

DNS Servers contain a wealth of data useful in the reconnaissance phase of an attack, including the hostnames and IP addresses of important resources. Malicious actors with an active foothold can leverage internal DNS to map large portions of target networks by performing sequential reverse DNS lookups—reversing internal IP addresses to hostnames. These hostnames often include useful naming schemas such as dc.acme.local that allow attackers to focus their attacks on the most useful corporate targets.



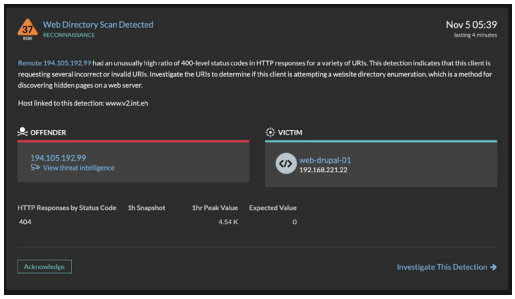
UDP Port Scanning

Attack tools such as NMAP allow attackers to scan CIDR ranges for a wide variety of open host and firewall ports. This provides attackers with information beneficial to the crafting of command and control channels, data exfiltration paths, and as an indication of which ports may yield additional attack surface.

APPENDIX B

Examples of Detections in Reveal(x)

Reconnaissance (Continued)



Web Directory Scanning

Web directory scans are a common means of reconnaissance. Vulnerabilities exist in every web hosting framework, from incorrectly configured permissions to failure to sanitize form field data inputs. The situation is complicated by the fact that organizations are slow to patch these servers due to their critical nature. Web directory scans allow attackers to determine what web framework an organization is utilizing while simultaneously testing for a variety of vulnerabilities and common security misconfigurations.

Additional issues exist due to the likelihood that organizations are either not collecting web log data, or are not leveraging the data for security purposes. Reveal(x) aids security teams by not only identifying potential attacks but also providing detailed information related to what URLs were scanned and the error codes associated with responses. This helps security teams identify weaknesses that attackers are attempting to exploit and rapidly remediate any security concerns.

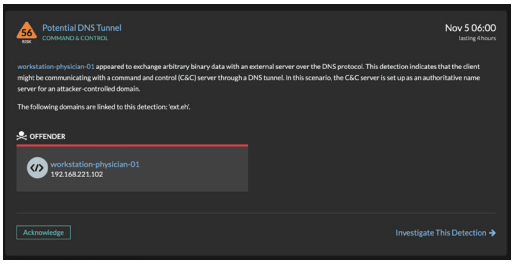
APPENDIX B

Examples of Detections in Reveal(x)

Command & Control

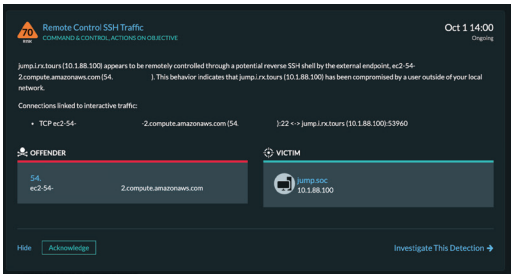
Unless the attacker has physical access to the devices on a network, they must control the systems remotely over the network. This command-and-control activity can take place over control protocols such as RDP, SSH, or telnet; over a custom protocol; or it may be disguised within the misuse of another protocol, such as DNS.

In addition to matching traffic with malicious IP addresses and domains contained in threat intelligence feeds, Reveal(x) uses behavioral machine learning models to recognize traffic patterns associated with common command and control activities.



Potential DNS Tunnel

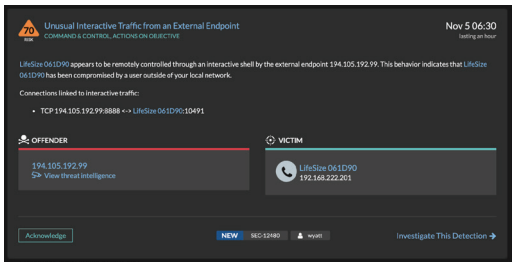
A DNS packet contains a number of different fields that attackers can use in creative ways, including command and control. By configuring malware to respond to specific bits, IP addresses, URIs, et cetera,. DNS can be used to control large botnets or individual machines and is also leveraged as a low bandwidth means of data exfiltration.



Reverse SSH

Reverse SSH is a communications technique used by a wide variety of malicious tools. Reverse SSH can be thought of as a code or script that calls back to a remote server, establishing an SSH connection and allowing a remote user access to a device. Hosts are often infected by malicious email attachments, downloads, or hijacked websites.

While this type of communication is occasionally used by legitimate vendors for a variety of purposes such as license key validation, the technique is not typically legitimate. Usually, reverse SSH is one of the first steps in establishing a foothold in a target environment.



Unusual Interactive Traffic from an External Endpoint

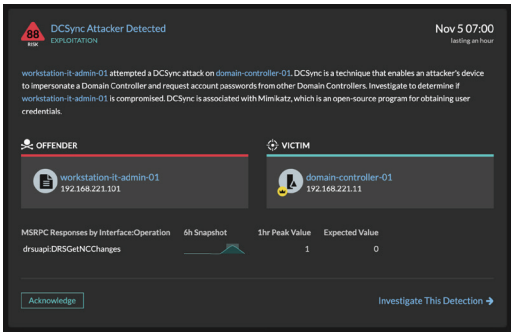
There are a wide variety of methods used to tunnel command and control traffic. Often these methods utilize protocols that already exist in a network in order to hide in plain site. Reveal(x) leverages its ability to ingest real-time traffic in high-bandwidth environments to monitor both encrypted and unencrypted protocols for evidence of command and control behavior. This monitoring looks for more than just the traffic patterns detectable with Encrypted Traffic Analysis, but also for plain text and encoded strings that are indicative of a remote shell.

APPENDIX B

Examples of Detections in Reveal(x)

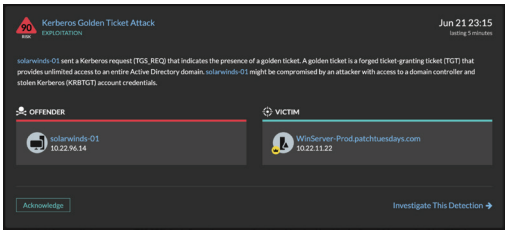
Exploit

Once they have identified their targets inside the network, attackers use tools like Cobalt Strike, Metasploit, and PowerShell Empire (to name a few) to gain access to sensitive systems by exploiting vulnerabilities. Detection at this stage of the attack lifecycle is crucial because, at this point, attackers still have not accessed sensitive systems. If defenders can detect and respond to attack activity at this stage, they can avoid financial and brand damage.



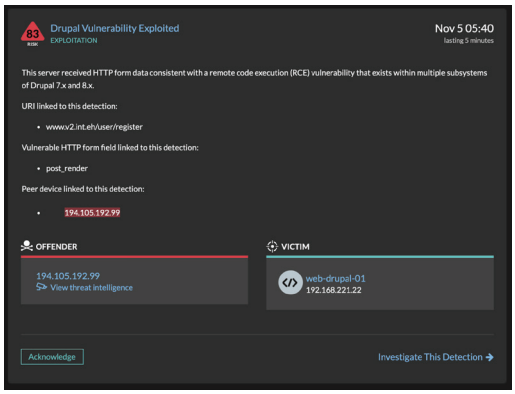
DCSync Attacker Detected

A DCSync attack facilitates access to corporate networks without the need to leverage additional code or log onto a domain controller. This is done by taking advantage of the necessity of domain controllers to replicate with one another. An attacker with proper privileges can impersonate a domain controller and trigger replication of the Active Directory database, gaining access to user accounts that can be used for additional post-exploitation attacks. Reveal(x) detects this style of attack by looking for unusual DCSync behavior and alerting SecOps personnel.



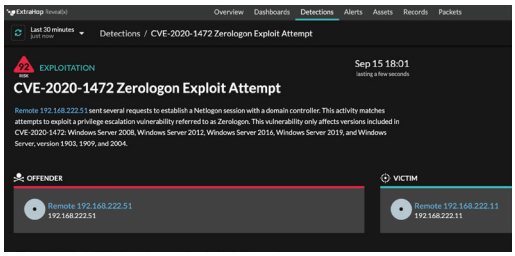
Kerberos Golden and Silver Ticket Attacks

Often closely linked with the presence of the Mimikatz malware, a Kerberos ticket attack involves an adversary gaining control over an Active Directory Key Distribution Service Account (also called KRBTGT). This account is then used to forge valid Kerberos Ticket Granting Tickets (TGTs), providing the attacker access to any resource or service on an Active Directory Domain and allowing attackers to reside on networks indefinitely by disguising themselves as credentialed administrator-level users.



Drupal Vulnerability Exploited

Drupal, like all web content frameworks, is vulnerable to a variety of attacks. Patches are continually released to address these vulnerabilities, however deployment of patches by sysadmins is often spotty due to change management windows. These vulnerabilities, when exploited, can provide an attacker with an array of capabilities up-to and including remote code execution (RCE). Reveal(x) evaluates incoming web requests for potential exploitation of these types of vulnerabilities. Detections provide forensic detail regarding the attack source and what methods of attack were attempted.



Zerologon

Zerologon allows attackers to use publicly available code to obtain full administrator privileges on Active Directory systems. Public proof-of-concepts leverage password change functionality to reset the domain controller machine account password. Since the account password is set by an adversary to a known value, an attack—like a DCSync for example—can be conducted to replicate ticket and service credentials, allowing for unfettered access to services and data throughout the organization.

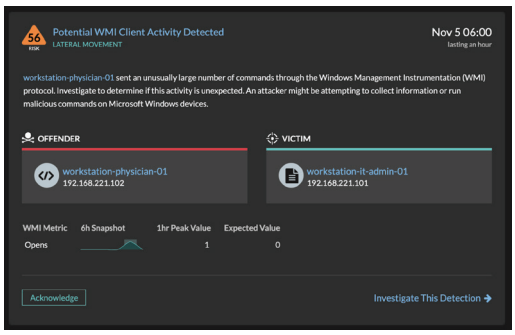
APPENDIX B

Examples of Detections in Reveal(x)

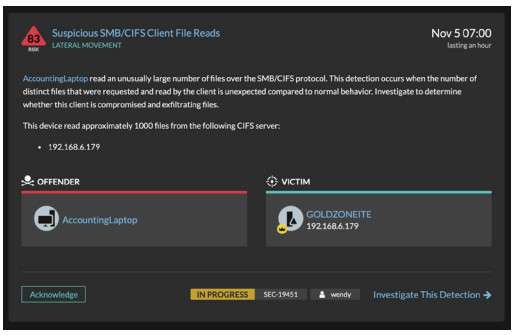
Lateral Movement

Attackers are opportunistic and enter the network however they can. Once inside, they need to move laterally to gain remote access to their target systems. Lateral movement detections focus on activities and behaviors that allow attackers to expand their presence from one system to another, escalate privileges, and collect data. Insider threats may also begin their mission at this stage of the attack lifecycle because they already have knowledge of the system and access, but lack the privileges needed to modify or steal data.

Reveal(x) uses machine learning to detect privilege escalation, exploits used to gain access to remote systems within the network, and unusual data movement as attackers collect data before exfiltration.



Potential WMI Client Activity
Windows Management Instrumentation (WMI) is commonly used by attackers for a variety of purposes, including authenticated code execution on remote Windows machines, data theft, persistence, and more. By creating a baseline of workstation behavior and monitoring WMI behavior in particular, Reveal(x) provides high-fidelity detections for suspicious WMI activity. Reveal(x) provides packet-level forensic data for each instance found and maps these instances into campaigns to aid security personnel in scoping an incident.

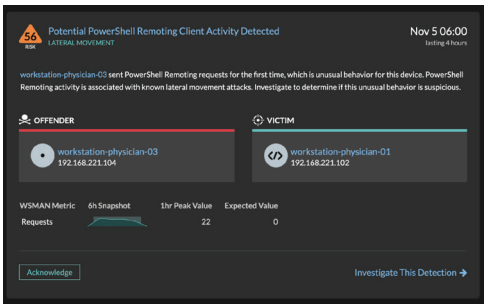


Suspicious SMB/CIFS File Reads
Security incidents frequently involve the theft or encryption of sensitive data located on remote file servers. Attackers also weaponize existing files in order to move laterally through a network. Reveal(x) monitors SMB/CIFS protocols for unusual behavior that is often indicative of data exfiltration campaigns and crypto-malware behavior. It gives SecOps personnel detailed information as to which files were tampered with, accessed, or placed on file server drives. This streamlines the investigation and response process associated with any incident.

APPENDIX B

Examples of Detections in Reveal(x)

Lateral Movement (Continued)



Potential Powershell Remoting Activity Detected

PowerShell is one of the most powerful tools available to Windows administrators and malicious actors alike. PowerShell can interface directly with Windows operating systems while simultaneously providing an intuitive scripting language to build automation on. Attackers often leverage PowerShell to maintain access to compromised endpoints (to be used as command and control nodes), to remotely control compromised endpoints, and a wide variety of other tasks.

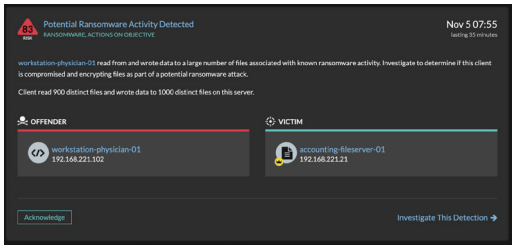
Reveal(x) monitors Windows endpoints for unusual PowerShell-related network activity and parses the activity it ingests for indications of malicious activity. In this way Reveal(x) is able to provide SecOps personnel with early detection capabilities when an endpoint is being remotely accessed via PowerShell.

APPENDIX B

Examples of Detections in Reveal(x)

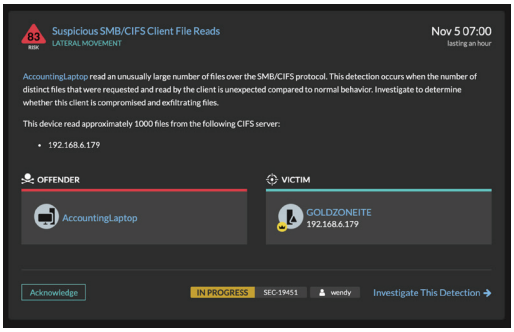
Actions on Objectives

Attackers may have various objectives, including sabotage, espionage, fraud, theft of intellectual property, theft of personal information, holding files for ransom, or simply making money by installing malware such as cryptomining or running botnets. Reveal(x) uses several methods to detect actions on objectives, including ransomware.



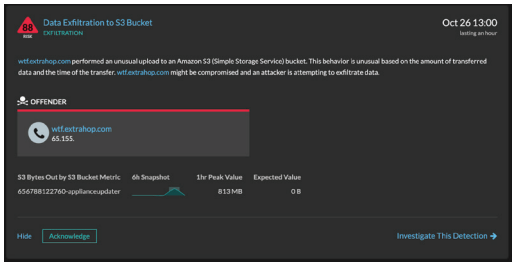
Potential Ransomware Activity

Modern ransomware works by encrypting a user's files, either locally or on remote file shares, then sending the decryption key to a central server. In this way attackers can be sure that the potential for decryption without paying the ransom is extremely low to non-existent. Reveal(x) monitors network activity for indicators that ransomware is encrypting files on remote file shares or beaconing back to command and control servers. In some cases it is even possible to extract the encryption keys from stored network traffic in order to restore files.



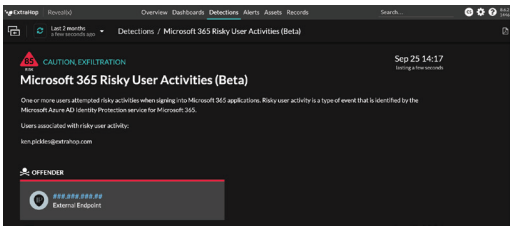
Potential SMB/CIFS Data Staging

During attack campaigns, attackers will often attempt to exfiltrate data from a target network. In order to do this efficiently, the data is first copied to a compromised endpoint and compressed for upload. Reveal(x) monitors file server access for unusual patterns of moving or copying behavior, allowing for rapid detection of potential data staging operations. This gives SecOps personnel the time needed to respond before the data leaves the environment.



Unexpected S3 Bucket Exfiltration

Malicious actors are often looking to do more than achieve a quick payday. Often they are also looking for intellectual property or confidential information. Attackers typically must find a way to extract copies of their target data from an organization's network. AWS S3 buckets are a common location for data exfiltration, as these servers are often used for legitimate internal operations. To address this attack, Reveal(x) will differentiate between standard S3 traffic and abnormal S3 traffic, providing the visibility needed to ensure an organization's data remains secure.



SaaS Detections: Microsoft 365 Suspicious Sign-In

The Reveal(x) Microsoft Office 365 Suspicious Sign-in detection is an example of authentication based detections. Malicious actors will often try to reuse credentials found on endpoints. This detection will assist security teams in identifying potentially compromised accounts and endpoints.