SANS

A SANS Product Review

ExtraHop Reveal(x) Expands Attack Investigations to Cover All Vectors

Written by **Dave Shackleford**

October 2019

Sponsored by:

ExtraHop

Introduction

In the past decade, the information security industry has learned a lot about what attackers do during campaigns against targets. While we don't always understand the motivation behind the attacks, most attacker goals focus on data access and exfiltration of sensitive data. ExtraHop's Reveal(x) security analytics product helps solve the challenge of security monitoring and response by providing security analysts with a platform that can rapidly analyze huge quantities of data without having to store full network packets.

Sophisticated attackers often use advanced malware-based espionage that can aggressively pursue and compromise specific targets. These attacks include social engineering tactics, such as spear phishing attempts. Once a compromise occurs, attackers attempt to maintain a persistent presence within the victim's network, escalating privileges and moving laterally within that network to extract sensitive information to locations under the attacker's control.

Various industry models describe attack phases. MITRE's ATT&CK is one such mature attack lifecycle providing a roadmap of stages to illustrate what security analysts are up against when dealing with attackers. The ATT&CK model includes the following stages:

- **Persistence—**Attackers set up backdoors and methods to retain access over time on the system.
- **Privilege escalation—**Attackers turn to DLL injection, or use setuid, privileged account access and more, with the intention of elevating privileges on the local system to gain more thorough control.

- **Defense evasion—**In this stage attackers use evasion attempts to avoid host defenses, among them intrusion detection, malware prevention and logging, including clearing shell history and logs, token manipulation and obfuscating files.
- **Credential access—**This stage is a mix of classic account attacks including brute force attacks against usernames and passwords, sniffing, private key compromise and dumping credentials from memory that can assist attackers in gaining access to new systems or furthering access in existing systems or applications.
- **Discovery—**Attackers look for other types of information they can leverage. This may include user data, privileges, devices, applications, services and data.
- **Lateral movement—**At this phase, attackers look to migrate from one compromised host to others in the environment. Techniques employed here may include "pass the hash" with credentials, remote admin and access tools, remote services and logon scripts.
- **Execution—**Attackers use various tools or methods to gain additional access in the environment, often leveraging tools like PowerShell, scripts, service-based vulnerabilities and many more.
- **Collection—**Attackers invariably want to collect data from compromised systems, which may include such data as clipboard info, input from the keyboard and other devices or screen/video captures.
- **Exfiltration—**Attackers interested in compromise for profit, as well as those with very specific goals, will always try to exfiltrate data from the environment. This may involve encrypting the data, setting up different types of network channels and protocols for moving data out of the network or scheduling data transfers.
- **Command and control—**For longer-term attack campaigns, attackers will seek "always on" control over compromised systems. Establishing a command and control (C2) mechanism on these hosts may involve things like custom protocols, encapsulated and tunneled content and the use of encryption.

Enterprise security teams have struggled to keep pace with attacker tactics and techniques, and many of the security tools we've relied on have not kept up with new methods of ingress, data access and exfiltration, either. Security teams are facing pressure to detect attacks and respond to them more rapidly, which is difficult when trying to find evidence of lateral movement, reconnaissance, privilege escalation and other stealthy behavior. There are many reasons why detection of attackers' traffic is so difficult. Among them:

- **Little telemetry available—**Most detection and monitoring platforms are not specifically oriented towards massive data collection and continuous monitoring based on machine learning.
- **No decryption—**Encrypted traffic can pose a big problem to security analysts trying to find evidence of malicious behavior.
- **Limited event logging—**Most organizations do not log and monitor many (or any) events from end user devices.

- **Organizational and data silos—**Many organizations are challenged to gain access to all the security and event data needed to see across the entire environment.
- **Traffic speed—**Currently, many internal networks operate at 40Gbps and may soon be sending network data at 100Gbps inside the data center. The resulting volume of data moving across the network can prove difficult to monitor, to say the least.

Reveal(x): Parsing Large Quantities of Data

SANS had an opportunity to take a second look at ExtraHop's Reveal(x) security analytics product. This is actually the second time we've reviewed Reveal(x), and the company has since added many enhancements and new features to augment the platform's behavioral-focused model of detection and response, all of which can help intrusion analysis and investigations teams analyze malicious behavior in their environments even more rapidly and effectively.

Reveal(x) solves many issues traditionally plaguing large-scale security analytics tools. We've identified five core security areas which Reveal(x) helps address:

- **Deployment model and flexibility—**One advantage of Reveal(x) is its out-of-band deployment model, which could make monitoring the environment more "stealthy" and in turn keep attackers from knowing they're being observed. Reveal(x) has support for both on-premises and cloud deployment in AWS and Azure, and also now has a new SaaS offering, Reveal(x) Cloud, eliminating the deployment and management overhead of the solution altogether. This new option takes advantage of the recently released AWS VPC Traffic Mirroring capability which can copy VPC network content in its entirety to a separate tool or location.
- **Broad visibility and context—**Reveal(x) reconstructs every transaction on the network and stores 4,800+ metrics for these transactions. This not only gives it excellent content for machine learning features, but also enables analysts to quickly understand the context of a detection and conduct ad hoc investigations since the metrics are all indexed and searchable. This can assist analysts in understanding the "blast radius" of an incident and what the attacker did previously on the network.
- **Machine learning—**To truly process massive quantities of data at scale and improve accuracy and insight over time, any enterprise security analytics platform should have demonstrable machine learning technology on the backend. While SANS did not explicitly test or delve into all aspects of Reveal(x)'s machine learning capabilities, they are well-documented and available for analysis and discussion. Unlike other network detection and response (NDR) solutions, Reveal(x) does not run its machine learning entirely on the installed appliance. The more compute-intensive models run in the cloud, and they carefully de-identify and tokenize and tokenizes all data to ensure GDPR compliance of the cloud detection service. This has the benefit of enabling Reveal(x) to run more than a hundred predictive models for each entity on the network, correlate detections across sensors for peer group analysis and apply rapid updates.

- **Depth and breadth of Layer 7 protocol analysis—**Reveal(x) has a deep application layer protocol analysis engine enabling the product to granularly inspect content and information contained in the transaction payload, such as methods, errors, SQL statements, DNS hostname lookups, file names, user names and the like.
- **Decryption—**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. (There are several methods available to accomplish this.)

The Reveal(x) engine performs automated asset discovery in the environment and then uses stream processing to auto-discover and classify every transaction, flow, session, device and asset detected. This entire process is passive and can perform at speeds up to 100Gbps. The benefit to this approach is twofold: It can help to immediately get a handle on system inventory and asset classification and it enables easily querying and exploring the entire inventory through the interface.

By emphasizing ease of use, deep analytics capabilities, built-in intelligence and search tools and rapid event triage, many SOC teams could hit the ground running quickly with Reveal(x). Our review environment was provided by ExtraHop and configured with a number of compromised systems configured to exhibit mock attack activity.

Reveal(x): Multiple Dashboards, Easy Navigation

As we made our way through Reveal(x), once again the graphic interface within the Reveal(x) console (the "Overview" section) stood out to us for its visual, friendly design. Reveal(x) has a very unique dashboard with a dynamic, ever-changing graphic overview of what is happening in the environment. This dashboard is a true network data visualization engine, which can be tuned to show specific time periods of events noted in the environment. In its most recent update, Reveal(x) now offers several different "big picture" dashboards showing specific things within the environment on three select tabs.

Figure 1. Security Overview Dashboard

The first tab, "Security," is similar in some ways to the core dashboard information in our last Reveal(x) review–we see a breakdown and summary of detections, which we can extend to weeks or months. This tab is aimed at a management audience in some ways, primarily because it provides a high-level snapshot of anything suspicious or malicious without too much granular specificity. The graphic visualizations have been completely updated, however, and are easy to understand at a glance, as seen in Figure 1.

This dashboard shows security events detected by categories (such as botnets or cryptomining), detections by device role (web servers, DNS servers, gateways, etc.) and top risk score detections with categorical listings (data exfiltration, DNS tunnels for command and control etc.). The second dashboard tab, "Network," provides a summary of what is going on in the network for various important protocols (LDAP, CIFS/SMB, DNS, etc.) and also "signal metrics" in the environment for the selected time period. This dashboard really shows off the in-depth visualizations Reveal(x) is capable of, with a continuously updated map of systems communicating, the protocols involved and the traffic quantity observed (see Figure 2).

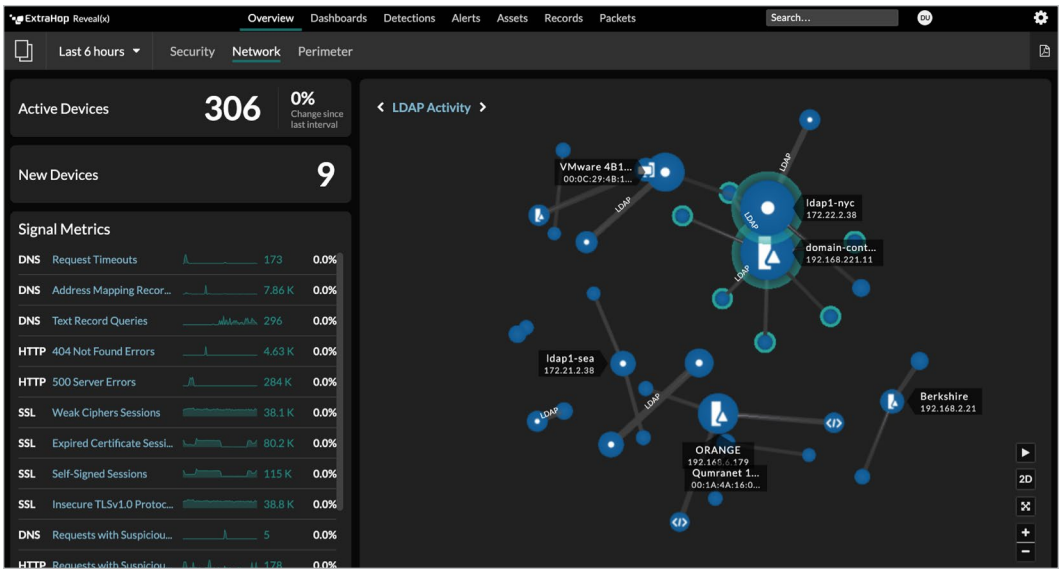

Figure 2. Network Overview Dashboard

This dashboard is highly interactive, and security and network analysts can easily select any observed traffic pattern or host, highlight what they want to see and start drilling down into more details. The final overview dashboard, the Perimeter, is a new feature added into the most recent versions of Reveal(x) and provides a summary view of North-South traffic including command and control and data exfiltration.

Whereas in our past review, we focused on Reveal(x)'s ability to detect lateral movement and East-West traffic internally between compromised systems, this time we broadened our scope to the new Perimeter feature. Perimeter adds powerful tools to watch for initial reconnaissance attempts; delivery of exploits and malware; data exfiltration; and attacker communications. To augment what's observed natively, Reveal(x) can ingest threat intelligence feeds in (Structured Threat Information Expression) STIX™ format so analysts can see matches on malicious domains and IPs noted in the wild. The dashboard shows total external traffic, suspicious inbound and outbound connections and highlights both exfiltration and C2 in visualizations (see Figure 3).

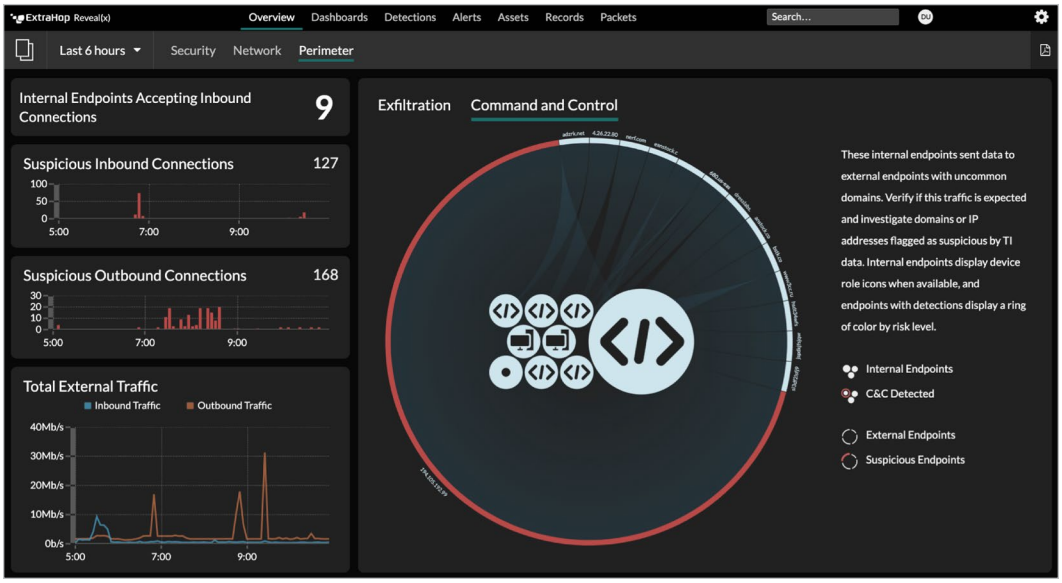

Figure 3. Perimeter Overview Dashboard

A number of built-in dashboards can show analysts extensive details about network traffic quantity and types, as well as in-depth activity with application traffic and individual systems listed out and system health (which shows a breakdown of device types detected in the environment and packet/traffic flow). A security dashboard specifically shows detections and alerts, threat intelligence data and a wide variety of hygiene data noted in the environment. For example, if Reveal(x) detects weak SSL ciphers in SSL (TLS) handshakes, failed DNS lookups or expired and self-signed SSL certificates, it will generate an event. The Security dashboard is shown in Figure 4.

Figure 4. Reveal(x) Security Dashboard

While we didn't spend too much time on this feature, it's worth noting analysts can easily customize their own dashboards in Reveal(x). They can generate new dashboards with many different data sources and chart types, and can create the layout by dragging and dropping elements on the design page.

Reveal(x)'s interface has a number of additional tabs in the console showing analysts very specific things as needed during investigations. We'll delve into those next.

Detections

The Detections tab shows all suspicious activity noted in the environment, and we can sort these detection events by detection time or by the highest risk score. Each of these can provide immediate links to more detail: records and even packet traces, threat intelligence insights where available, activity maps and the current state of the activity (for instance, ransomware activity still seen actively). Reveal(x) also displays associated ticketing information including ticket numbers, ticket status and who the ticket is assigned to through integration with ticketing and service management platforms like ServiceNow. See Figure 5.

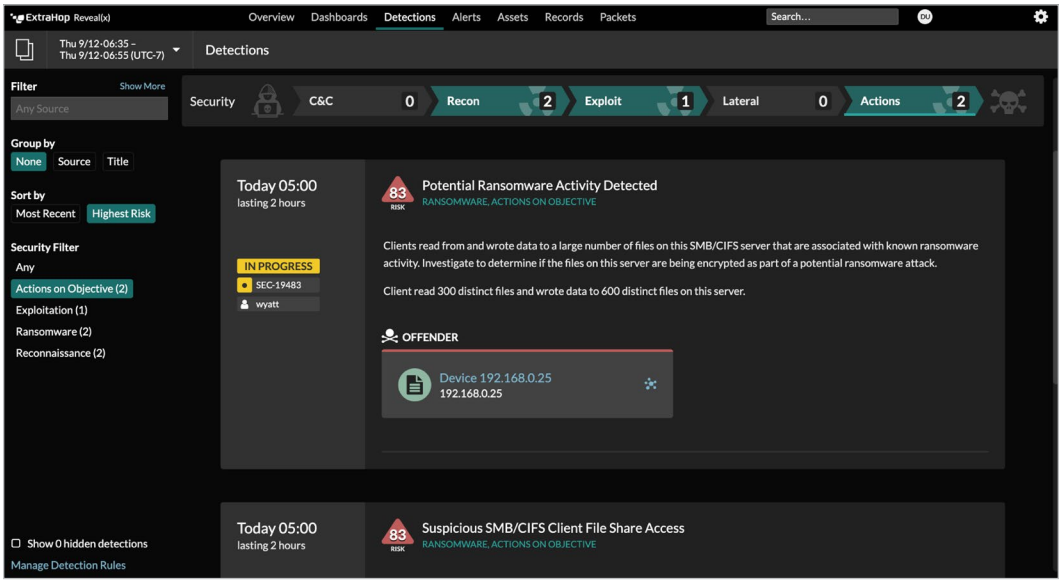

Figure 5. Reveal(x) Detections Tab

Assets

The Assets tab shows a quick breakdown of protocols and devices noted in the environment, along with what clients are connected (where appropriate) and any anomalies detected (which link back to the Detections view). This can be very useful for a number of use cases, including performance and traffic monitoring for both the network operations center (NOC) and security together, as well as digging quickly into who is talking to what in a very granular way. Figure 6 shows a breakdown of database methods seen in queries, top users and some performance data as well.

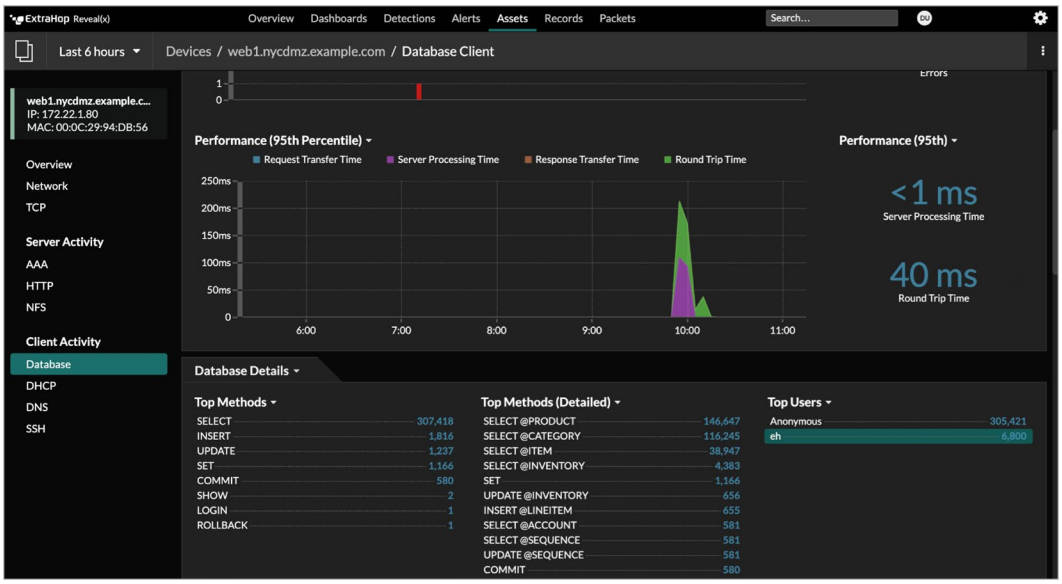

Figure 6. Assets Tab with Database Reporting

Records

The Records tab shows the actual transactions observed in the environment with much more granular detail. In this tab, analysts can query for such information as database transactions, SSL handshakes, SSH session information and DNS requests and responses. For example, we looked into the exact database transactions executed by one of the users seen in the Assets tab previously but with more detail and the ability to group by a large number of categories (database types, client ports, record types). See Figure 7.

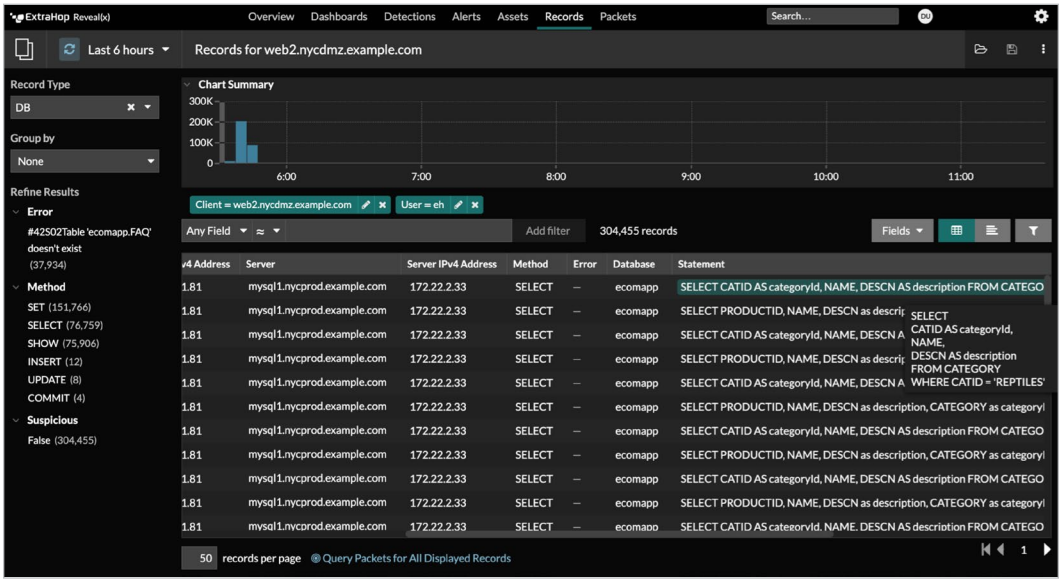

Figure 7. Database Records for a Specific User

Reveal(x) gathers and reports all of the content from these various tabs without the need for logs—rather, it passively observes everything on the wire. This is especially useful when robust logging may not be available. A good example is databases where extensive logging causes undesirable performance impacts.

Continuous Packet Capture

One fantastic feature in Reveal(x) was the Continuous Packet Capture view (the “Packets” tab on the top menu). Here we were able to select specific devices or time ranges and actively query all traffic at a raw packet level, even downloading a packet capture (PCAP) file we can use for offline analysis in Wireshark and other protocol analysis and packet inspection tools. This is a major differentiator for a network security analytics tool that could easily help both network and security teams troubleshoot problems or investigate incidents much more quickly and thoroughly. See Figure 8.

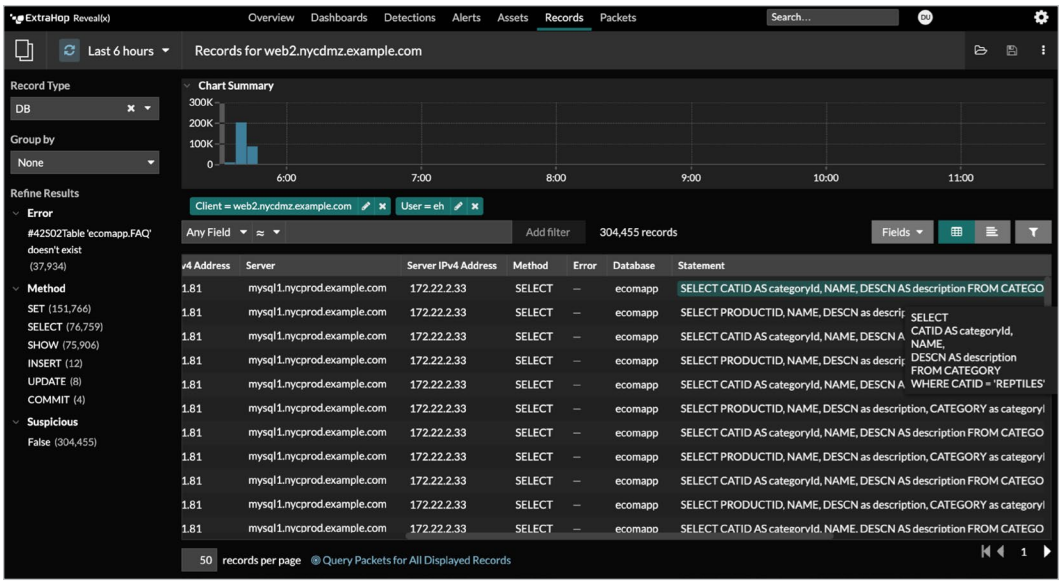

Figure 8. Packets Tab

Use Cases: Reveal(x) in the Real World

To look at Reveal(x) through the lens of a security analyst, we explored several real-world scenarios. These were intended to demonstrate how actual malicious traffic and attack attempts were detected and to show how we could respond and improve operational effectiveness with the product. We chose to focus on a few different, common areas in security operations: detection and response, threat hunting and security controls and “hygiene” in the environment.

Scenario 1: Detection/Response with Reveal(x)

The first use case we walked through with Reveal(x) focuses on detection of malicious activity and response investigations. In the new version of Reveal(x), detections are still primarily developed in a few ways: through machine learning models (including time-series analysis, behavior graph analysis and peer group behavior modeling) and rules-based detections that are automatically updated through the cloud. Customers can also create their own rules-based detections.

One detection modality we examined is Reveal(x)’s predictive models–which use the predictive machine learning models that run in ExtraHop’s cloud service we mentioned earlier. To start our investigation, we looked in the Detections tab for high risk scores associated with suspicious activity.

One relatively high-risk activity looked like a brute force credential attack against CIFS/SMB services. Unexpected behaviors observed in the environment indicate likely attack traffic. By clicking into the detection, we observed two examples of the predictive

models for the device "**web-drupal-01**" that helped Reveal(x) detect brute force attacks against this server, the CIFS Errors by Method and Error values shown in Figure 9. Of note: This application-layer detection required decryption, and Reveal(x) builds up to 100 models for every observed entity its cloud-based machine learning can use to improve detection accuracy.

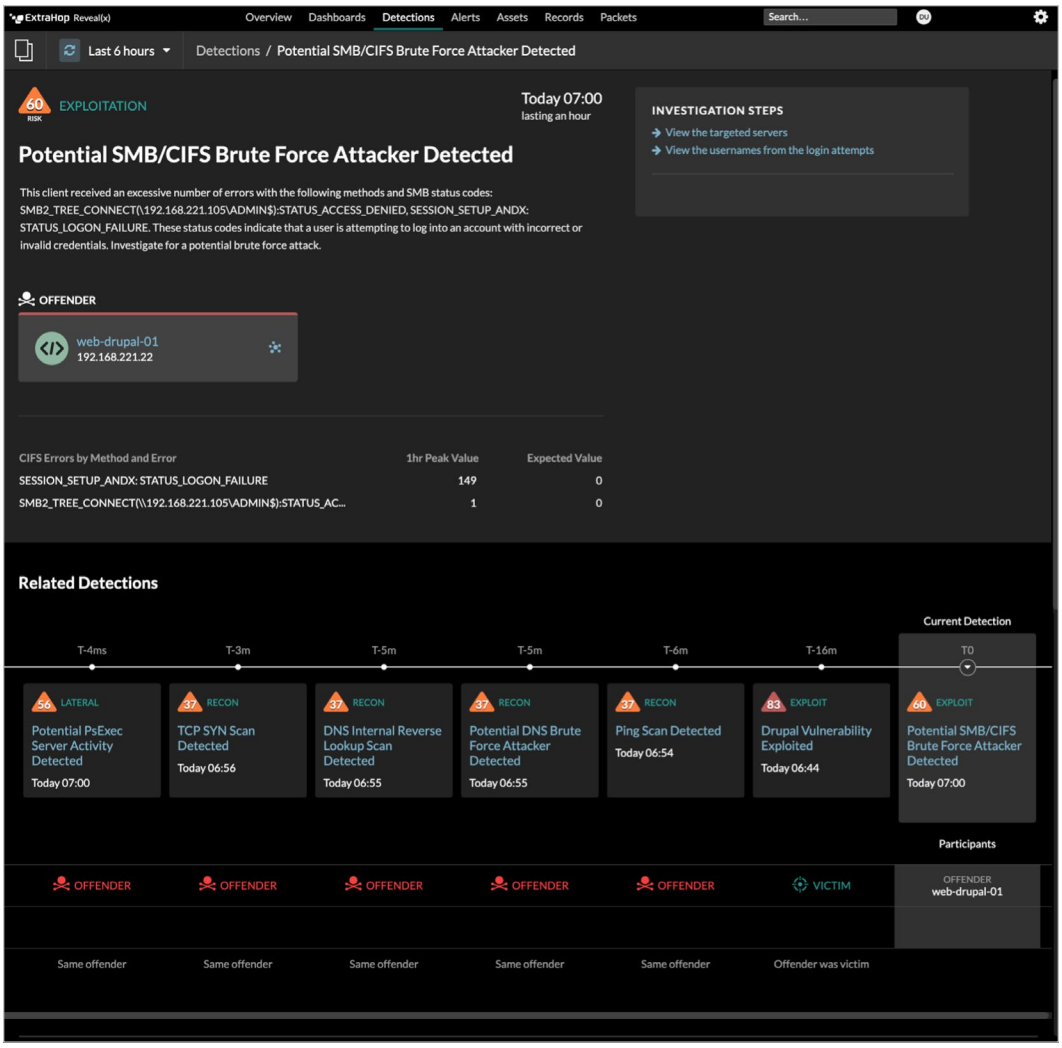

Figure 9. Reveal(x) Predictive Models for Brute Force Credential Attacks

Reveal(x) includes a wealth of context along with detections, such as expected range and deviation, devices involved, how they calculated the risk score, links to outside resources such as the CVE listing or MITRE ATT&CK tactics, techniques and procedures (TTPs) and next steps for investigators. These extra details and suggested next steps are what ExtraHop calls "guided investigation" and are meant to help junior analysts tackle more of the investigative process faster on their own. With a single click, we also broke out the full system details and could drill down to event information; this ability helps boost analyst productivity by giving them contextual information they would normally have to look for in system logs, which may not be readily available. Similarly, we performed full username lookups for the associated event with a single click, as shown in Figure 10.

Figure 10. Brute Force Usernames in Detect Event

Reveal(x) also focuses on privilege escalation and use of privileges, along with administrative tools and protocols across the network (see Figure 11).

Peer grouping is an underlying mechanism Reveal(x) uses to analyze behavior and gauge how like systems that behave similarly and send/receive similar protocols and traffic may suddenly act unusually. This set of monitoring algorithms aligns with the machine learning prediction and privilege use detection methods as well.

Another detection modality we examined was rules-based detection, which Reveal(x) can push down to platforms. These "triggers" in the rules function are much like traditional intrusion detection signatures that match patterns in observed traffic. ExtraHop releases many of these and pushes them to the Reveal(x) platform automatically, but analysts can create new rules or modify rules easily by simply updating the code for them in the Settings → Triggers section of the console (as seen in Figure 12).

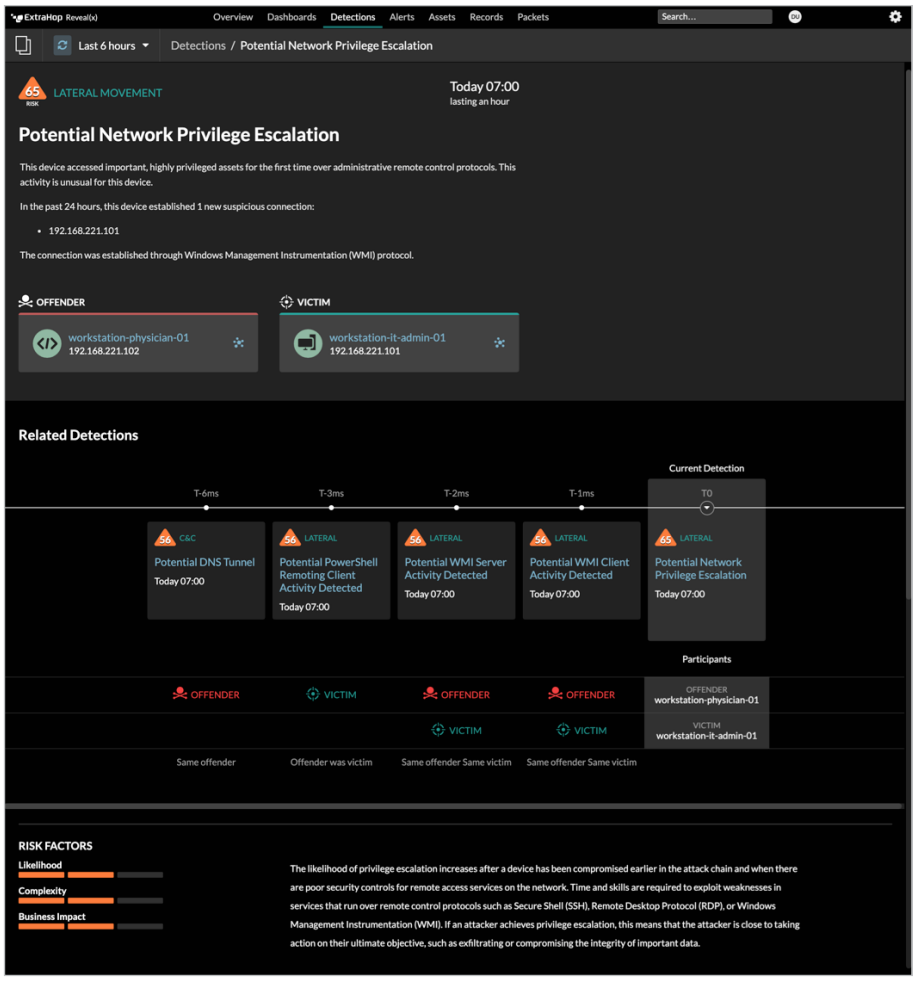

Figure 11. Example of Possible Privilege Abuse

In the newest Reveal(x) version, analysts can build their own full "detection cards" with next steps, links to other parts of the product and more.

To streamline or automate remediation, Reveal(x) integrates with ticketing, SOAR, NAC and NGFW systems to create tickets, kick off orchestrated workflows, block malicious traffic, quarantine devices or lock user accounts. The company has built a number of supported integrations, but also offers a REST API and other mechanisms to facilitate custom integrations.

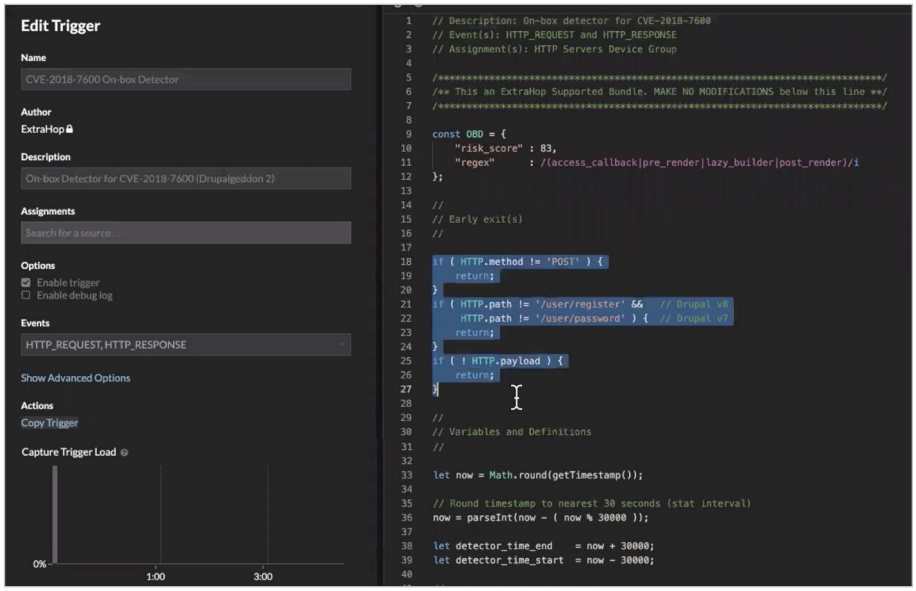

Figure 12. Modifying Rule Triggers

Scenario 2: Proactive Threat Hunting–Vulnerable Systems and Applications

Another primary use case for Reveal(x) is threat hunting, where security teams proactively look for indicators of compromise or possible vulnerabilities in the environment attackers are actively exploiting in the wild. One example we explored was

a case of a known Drupal vulnerability allowing for remote exploitation through a known HTTP query with the term “register” in it. On the Records tab, we switched to look for only HTTP traffic and included this word in the “URI” query filter (see Figure 13).

Reveal(x) returned two records (one GET and one POST). We can refine the query to only look for POST methods, as we know from the CVE that only POSTs are vulnerable. Drilling down into the POST details will show us additional information, including what system is communicating and where the client connection came from.

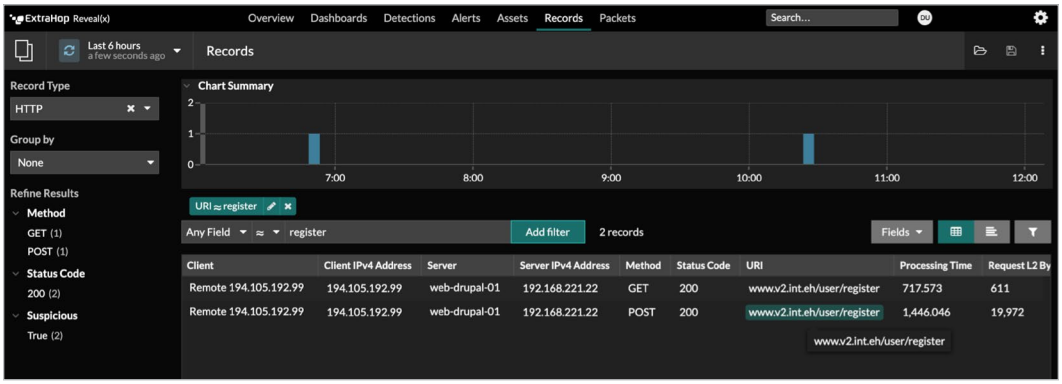

Figure 13. Querying for a Drupal Vulnerability

Again, as seen earlier, analysts can add or modify signature-based rules called “triggers” that can help to quickly add simple pattern matching methods for threat hunting as much as detection and response. Additionally, Reveal(x) detections and metrics can be streamed easily to SIEM platforms through ExtraHop partner integration, allowing analysts to view Reveal(x) data in their native SIEM consoles with links included to bring them back to Reveal(x) for continuing investigations.

Scenario 3: Hygiene and Compliance

Reveal(x) has a number of features teams could easily use to facilitate continuous monitoring and improvement of overall cybersecurity hygiene to both reduce attack surface and operational effort. The platform’s ability to easily monitor the environment for specific devices, applications in use, protocol behavior and traffic patterns could help organizations meet compliance and regulatory requirements as well as adhere to best practices like the CIS Critical Controls.

For example, Reveal(x) includes prebuilt dashboards looking for SSL/TLS certificate use along with cryptographic ciphers associated with the handshakes invoked. This is an excellent way for security analysts to periodically review the types of certificates deployed and used in the environment and lets them report on any weak ciphers that may be in use or expired as well as on any self-signed certificates not updated or replaced (see Figure 14).

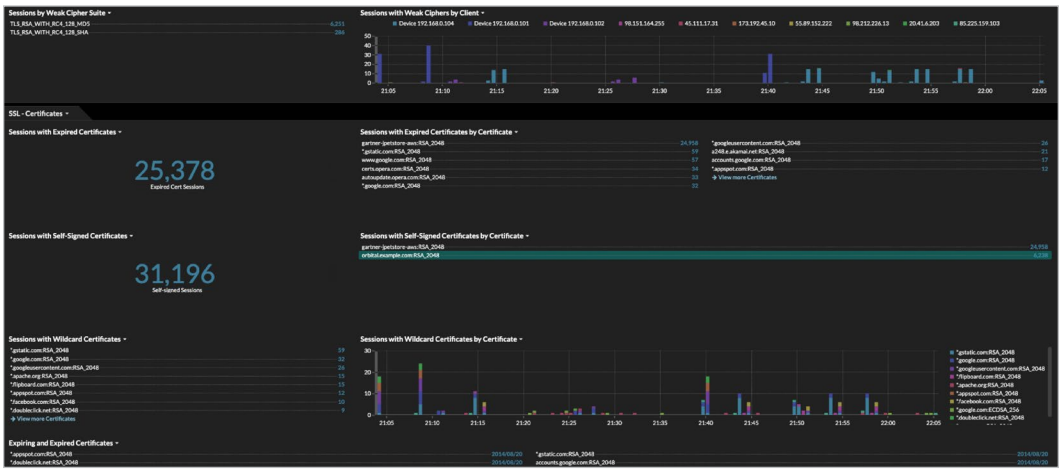

Figure 14. Weak Ciphers and Expiring Certificates

Analysts can also query for legacy protocols in use with Reveal(x). For example, we searched in the Assets tab for "CIFS Servers" and viewed Methods to see which SMB versions are in use. (SMBv1 is deprecated and has been exploited by numerous malware variants.) Similarly, we looked to see what devices are functioning as remote desktop protocol (RDP) servers and if they are communicating with external IPs (unfortunately, our test data set did not include RDP), as shown in Figure 15.

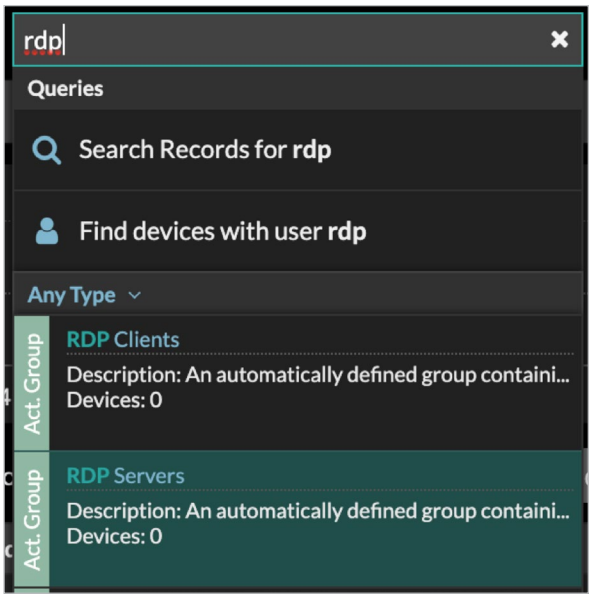

Figure 15. Querying for RDP Activity

Conclusion

After delving into the updated platform, we feel the latest ExtraHop Reveal(x) features have significantly enhanced the product since our first review. The tool is fast, thorough and provided an enormous range of options for searching and querying activity within the environment. As we stated before, the interface is very intuitive and easy to learn (and even better than before), making for a great security analysis platform for Tier 1 analysts all the way to senior investigators. The level of detection detail as well as the "next steps" and links between product areas for quick hunting activities, are some of the best we've seen.

We went into the original review with the core idea of looking into lateral movement and the vexing security issue of monitoring East-West traffic in environments. Reveal(x) still does this well, while also helping build an asset inventory and an accounting of protocols and applications in use. Now, the platform is also able to monitor inbound and outbound traffic in great detail, providing much more complete coverage of the network landscape. We didn't cover Reveal(x)'s integration with existing security tools organizations may have, but ExtraHop continues to bolster its usefulness through its very open integration ecosystem with partners in the SIEM, NGFW, ticketing and orchestration and automation categories. This open approach can significantly enhance the continuity of the security operations practice and facilitate improved automation and speed of detection to investigation.

Overall, we found Reveal(x) detailed and flexible for any range of security operations teams who need better visibility into network behavior in their environment—and it has the added benefits of deep investigation and hunting tools.

About the Author

Dave Shackleford, a SANS analyst, instructor, course author, GIAC technical director and member of the board of directors for the SANS Technology Institute, is the founder and principal consultant with Voodoo Security. He has consulted with hundreds of organizations in the areas of security, regulatory compliance, and network architecture and engineering. A VMware vExpert, Dave has extensive experience designing and configuring secure virtualized infrastructures. He previously worked as chief security officer for Configuresoft and CTO for the Center for Internet Security. Dave currently helps lead the Atlanta chapter of the Cloud Security Alliance.

Sponsor

SANS would like to thank this paper's sponsor.