A SANS Product Review

Investigate East-West Attacks on Critical Assets with Network Traffic Analysis

Written by **Dave Shackleford**

October 2018

Sponsored by:

ExtraHop Networks

Introduction

Today, an attacker's goals are focused on data access and exfiltration. To gain entry into an environment, sophisticated attackers aggressively pursue and compromise specific targets, often using social engineering tactics such as spearphishing. This attack activity is described by the Lockheed Martin Cyber Kill Chain[1] and is focused on initial reconnaissance and weaponization to exploit and control a device.

Once a compromise has occurred, 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. This post-compromise attack activity is described by another well-documented industry model, Mitre's ATT&CK™, which lays out an attack campaign and its phases. It focuses more on the specific internal mechanics of an attack beyond the reconnaissance and weaponization phases of the kill chain, and includes the following:[2]

- **Persistence.** Attackers set up backdoors and methods to retain access over time on the system.
- **Privilege escalation.** This is accomplished through DLL injection, using `setuid` and privileged account access, among other methods, with the intention of elevating privileges on the local system to gain more thorough control.

[1] "Deconstructing The Cyber Kill Chain," Dark Reading, Nov. 18, 2014, www.darkreading.com/attacks-breaches/deconstructing-the-cyber-kill-chain/a/d-id/1317542

[2] Mitre, Adversarial Tactics, Techniques & Common Knowledge, https://attack.mitre.org/wiki/Main_Page

- **Defense evasion:** Attackers use defense evasion attempts to avoid host defenses, such as intrusion detection, malware prevention, logging and more. Examples include clearing shell history and logs, manipulating tokens and obfuscating files.
- **Credential access:** Classic account attacks that include brute-force attacks against usernames and passwords, sniffing, private key compromise and the dumping of credentials from memory can assist attackers in gaining access to new systems or furthering access in existing systems or applications.
- **Discovery:** The discovery phase is when 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:** Execution is the stage where attackers use various tools or methods to gain additional access in the environment, often leveraging tools such as PowerShell, scripts, service-based vulnerabilities and many more.
- **Collection:** Attackers invariably want to collect data from compromised systems, which may include clipboard info, input from keyboard and other devices, screen/video captures and other such data.
- **Exfiltration:** Attackers interested in compromise for profit, as well as those with very specific goals, always look 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, and scheduling data transfer.
- **Command and control:** For longer-term attack campaigns, attackers seek "always on" control over compromised systems. Establishing a command-and-control mechanism on these hosts may involve custom protocols, encapsulated and tunneled content, encryption and more.

With such attack vectors identified and well defined, why are we not catching the attacks seen in the wild today? In short, while attacks and methods are constantly changing, our tools and approaches to fighting them have not evolved as quickly. Most tools generate event data based on signatures and rules, and most security event management and analytics tools are non-intuitive and challenging to configure, use and maintain. At the same time, 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.

All of these under-the-radar activities take place within the east-west corridor of the environment, hiding within the network traffic among devices, applications and data storage in the virtual perimeter. While organizations have many products that protect their endpoints and network perimeter, visibility inside the environment is generally a challenge. We've identified five reasons why the detection of lateral movement and attacker's east-west traffic can be so difficult in Table 1.

With this context in mind, SANS reviewed ExtraHop Network's Reveal(x) network traffic analysis platform. Reveal(x) offers security analysts a way to rapidly make sense of huge quantities of data without having to store and query full network packets, which helps address the issues of limited logging and poor or no telemetry, as described in Table 1. It also provides a more behavior-based model of attack detection and response prioritized according to a dynamic inventory of critical assets, so intrusion analysis and investigations teams can more rapidly and efficiently detect and investigate malicious behavior in their environments. The data and analytics can easily be leveraged by numerous teams within an organization and can be integrated in security and ticketing systems. Finally, Reveal(x) has the ability to decrypt SSL/TLS-protected traffic, and even has a solution for traffic protected by Perfect Forward Secrecy (PFS) ciphersuites. By emphasizing ease of use, deep analytics capabilities, unsupervised machine learning, integrated and natural search tools, and rapid event triage, Reveal(x) is a product with which many security operations center (SOC) teams could hit the ground running. Our review environment was provided by ExtraHop, and was configured with a number of systems that had been compromised and configured to exhibit mock attack activity.

Table 1. Reasons Lateral Movement and Attackers' East-West Traffic Are Difficult to Detect

Reason	Description
Limited telemetry	Companies have few tools providing visibility into east-west traffic. Most detection and monitoring platforms are not specifically oriented toward massive data collection. Instead, such platforms generate log data, which collects data limited to the intention and capability of the tool when the log was defined. Usually, any monitoring is scheduled or periodic, rather than continuous, leaving holes in the data captured.
No decryption	Encrypted traffic can pose a big problem to security analysts trying to find evidence of malicious behavior.
Limited logging	Logs can be challenging to collect at scale, and systems such as databases are not logged because of server performance concerns.[3] Moreover, most organizations do not log and monitor many (or any) events from end-user devices. In addition, savvy attackers may delete or modify logs after compromising a system.
Organizational and data silos	Many security 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. This can prove difficult to monitor, to say the least. Tools designed for lower capacity will drop traffic or reduce analysis, obscuring attack activities.

[3] "Setting Up a Database Security Logging and Monitoring Program," August 2018, www.sans.org/reading-room/whitepapers/application/setting-database-security-logging-monitoring-program-34222

Browsing the Interface

Reveal(x) has a friendly graphical interface, with a unique initial landing page that shows a dynamic, real-time graphic overview of what is happening in the environment. This dashboard is a true network data visualization engine that can be tuned to show specific time periods of events noted in the environment. Figure 1 illustrates the last six hours of events and calls immediate attention to high-risk detections. Reveal(x) can display up to 30 days of analysis on a single appliance (this can be extended with supplemental network-attached storage).

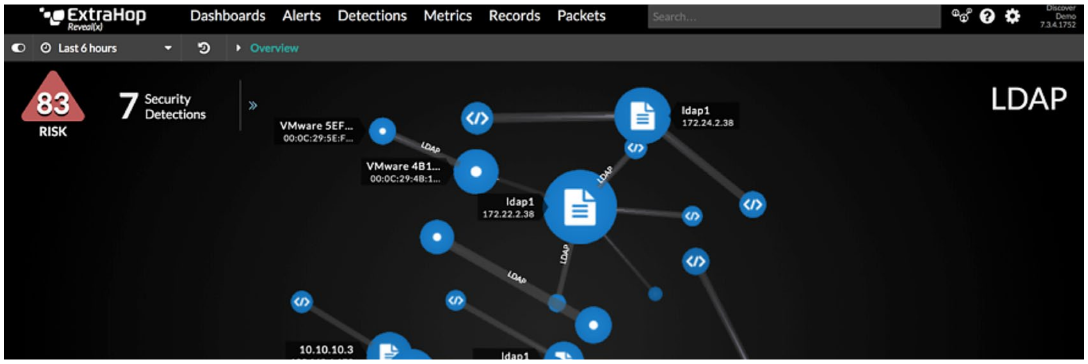
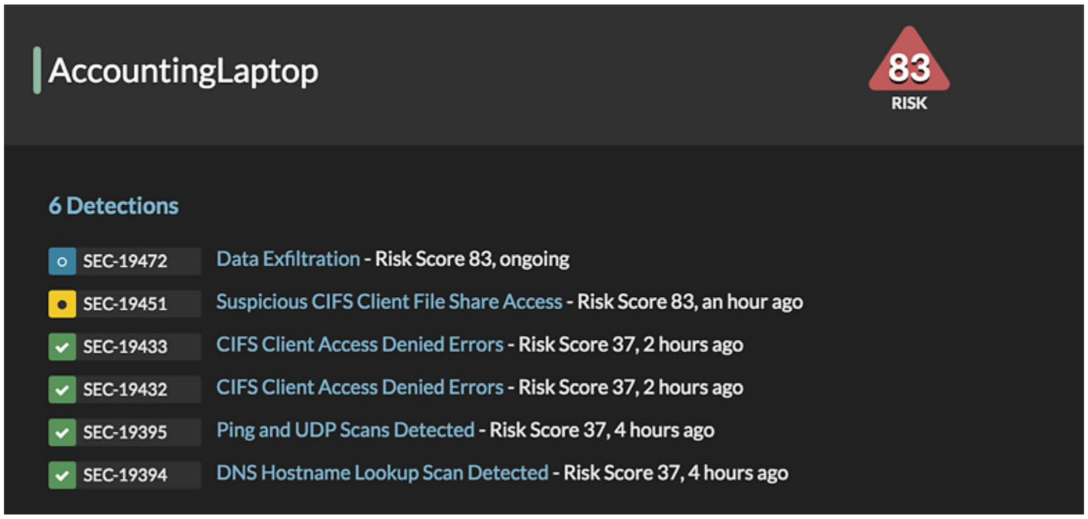


Figure 1. Event Overview Dashboard

By clicking the *Security Detections* count in the upper-left corner, an analyst can quickly bring out a more detailed pane that shows the specific events and behaviors noted—each of which includes an initial risk score, event details, the status of the ticket and a link to drill in deeper immediately (see Figure 2).

This pane alone is useful as a starting point for triage to home in on what is most critical. Because Reveal(x) does not rely on rules and signatures, it can present SOC analysts with only the anomalies and events that are worth investigating, without the false positives that overwhelm many SecOps teams. SOC analysts will still need to apply their knowledge of the environment to prioritize the events presented. But when something does warrant additional investigation, Reveal(x) makes it easy to drill down into live activity maps and device activity.



AccountingLaptop

83
RISK

6 Detections

○	SEC-19472	Data Exfiltration - Risk Score 83, ongoing
●	SEC-19451	Suspicious CIFS Client File Share Access - Risk Score 83, an hour ago
✓	SEC-19433	CIFS Client Access Denied Errors - Risk Score 37, 2 hours ago
✓	SEC-19432	CIFS Client Access Denied Errors - Risk Score 37, 2 hours ago
✓	SEC-19395	Ping and UDP Scans Detected - Risk Score 37, 4 hours ago
✓	SEC-19394	DNS Hostname Lookup Scan Detected - Risk Score 37, 4 hours ago

Figure 2. Additional Security Detection Details

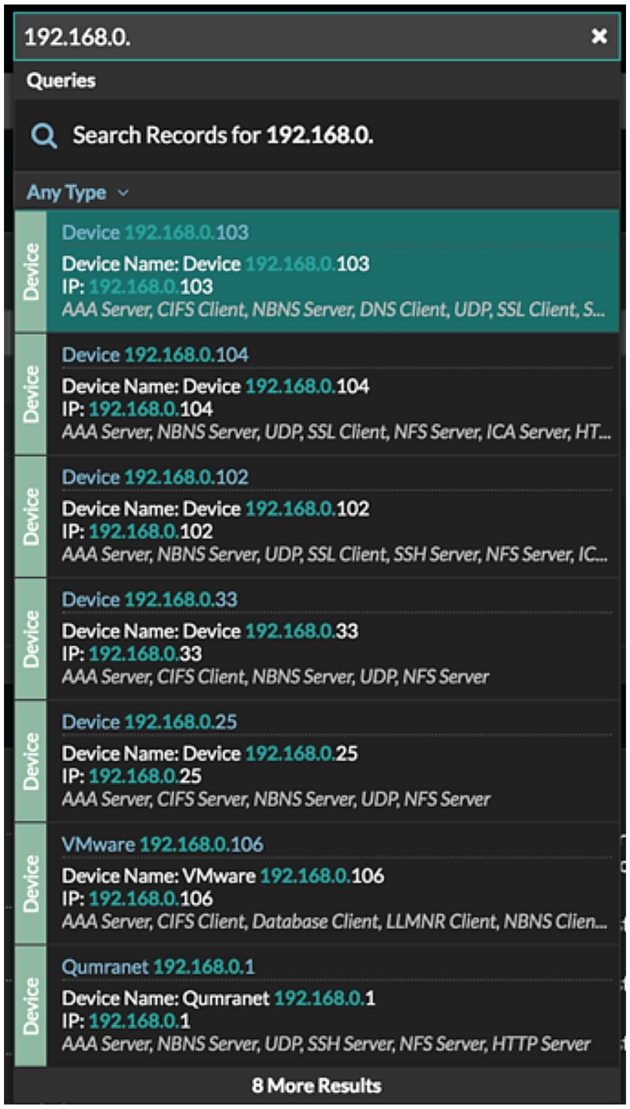
It's worth noting that ExtraHop builds this security event dataset through the use of machine learning. Reveal(x) collects enormous quantities of network data, decrypts and processes the data through its analytics engine, and runs the data through the company's cloud-based machine-learning environment. This service ingests lightweight metric information from on-premises appliances (which produce network metadata) and spots unusual patterns of protocol and application traffic that indicate unusual activity.

Network Discovery

Reveal(x) also excels at supporting network discovery, which is key to any security model. The first of the Center for Internet Security (CIS) Critical Security Controls[4] is entirely focused on shoring up organizations' common lack of visibility within in-house and cloud network environments through maintaining a sound inventory of systems. The security concept "you can't secure what you don't know about" holds true here, and this control (network inventory maintenance for both authorized and unauthorized hosts) has been the highest-priority control since the list's inception.

Reveal(x) 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. For example, if a system is responding to DNS requests, then Reveal(x) classifies it as a DNS server. This entire process is passive (out-of-band), and can perform at speeds up to 100Gbps—which can help an analyst quickly get a handle on system inventory and asset classification and helps solve some of the classic problems of lateral movement detection in busy networks described earlier. Moreover, Reveal(x) is able to prioritize critical assets such as databases, file servers and Active Directory infrastructure based on rules set by the user.

The entire inventory can even be queried and explored easily through the Reveal(x) interface. Figure 3 shows a global search being composed for an IP address, with preview results shown. The devices in the search results are tagged with their classifications, such as "AAA Server," "SSH Server," "LLMNR Client" and so forth.



Device 192.168.0.103
Device Name: Device 192.168.0.103
IP: 192.168.0.103
AAA Server, CIFS Client, NBNS Server, DNS Client, UDP, SSL Client, S...

Device 192.168.0.104
Device Name: Device 192.168.0.104
IP: 192.168.0.104
AAA Server, NBNS Server, UDP, SSL Client, NFS Server, ICA Server, HT...

Device 192.168.0.102
Device Name: Device 192.168.0.102
IP: 192.168.0.102
AAA Server, NBNS Server, UDP, SSL Client, SSH Server, NFS Server, IC...

Device 192.168.0.33
Device Name: Device 192.168.0.33
IP: 192.168.0.33
AAA Server, CIFS Client, NBNS Server, UDP, NFS Server

Device 192.168.0.25
Device Name: Device 192.168.0.25
IP: 192.168.0.25
AAA Server, CIFS Server, NBNS Server, UDP, NFS Server

VMware 192.168.0.106
Device Name: VMware 192.168.0.106
IP: 192.168.0.106
AAA Server, CIFS Client, Database Client, LLMNR Client, NBNS Clien...

Qumranet 192.168.0.1
Device Name: Qumranet 192.168.0.1
IP: 192.168.0.1
AAA Server, NBNS Server, UDP, SSH Server, NFS Server, HTTP Server

Figure 3. Global Search with Role-Based Device Tagging

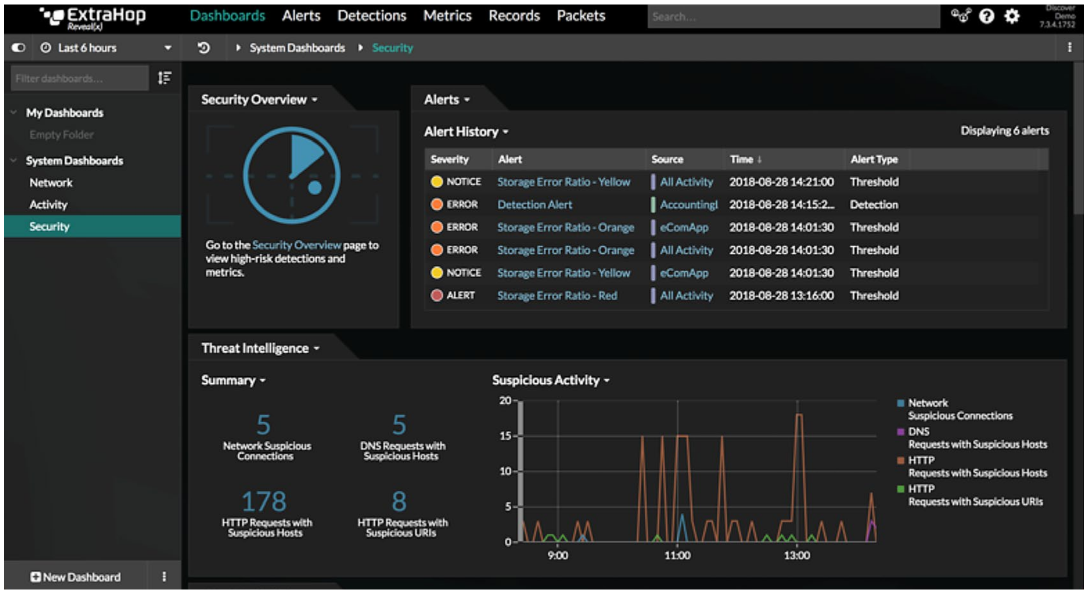
Dashboards

After exploring the initial landing page, we looked at the three additional dashboards Reveal(x) includes that may prove useful for different users of the platform. The Network Activity dashboard includes information and metrics about network throughput, packet and frame types seen on the wire, latency and connections. The Activity dashboard shows a mix of information related to network traffic, security alerts and events, and it provides protocol/application breakdowns for that traffic noted. The Security dashboard breaks down the security alerts—which are a summary of suspicious activities identified by ExtraHop's machine learning–driven detection—as well as matches with threat

[4] "CIS Critical Security Controls," www.cisecurity.org/controls

intelligence feeds, and provides specific details on SSL/TLS and DNS traffic (covered later). The Security dashboard is shown in Figure 4.

Reveal(x) also enabled us to drill into security events through the Alerts and Detections screens, which will be covered in the "Scenario 1: Breach Detection and Response" section. At its core, Reveal(x) does analysis in memory based on full-stream reassembly of packets observed on the wire. It stores the Layer 2 through Layer 7 metadata about transactions, then writes packets afterward as an option. This approach does not rely on full-packet capture to assess and monitor the environment, which is a true differentiator from most network monitoring and forensics platforms. For those customers that may need the full-packet capture data for more thorough network forensics, however, this is available in fully indexed form in the Packets pane.



Go to the Security Overview page to view high-risk detections and metrics.

Severity	Alert	Source	Time ↓	Alert Type
NOTICE	Storage Error Ratio - Yellow	All Activity	2018-08-28 14:21:00	Threshold
ERROR	Detection Alert	Accountingl	2018-08-28 14:15:2...	Detection
ERROR	Storage Error Ratio - Orange	eComApp	2018-08-28 14:01:30	Threshold
ERROR	Storage Error Ratio - Orange	All Activity	2018-08-28 14:01:30	Threshold
NOTICE	Storage Error Ratio - Yellow	eComApp	2018-08-28 14:01:30	Threshold
ALERT	Storage Error Ratio - Red	All Activity	2018-08-28 13:16:00	Threshold

5	Network Suspicious Connections
5	DNS Requests with Suspicious Hosts
178	HTTP Requests with Suspicious Hosts
8	HTTP Requests with Suspicious URIs

Figure 4. Reveal(x) Security Dashboard

Ultimately, the entire Reveal(x) environment was incredibly simple to navigate. Any SOC analyst, regardless of experience level, could quickly employ Reveal(x) to find pertinent information in a security investigation, drilling down to the level of detail desired with just a click or two.

Getting More

While we won't spend too much time on this feature, it's worth noting that analysts can easily customize their Reveal(x) dashboards with many different data sources, chart types and layouts by dragging and dropping them onto the design page.

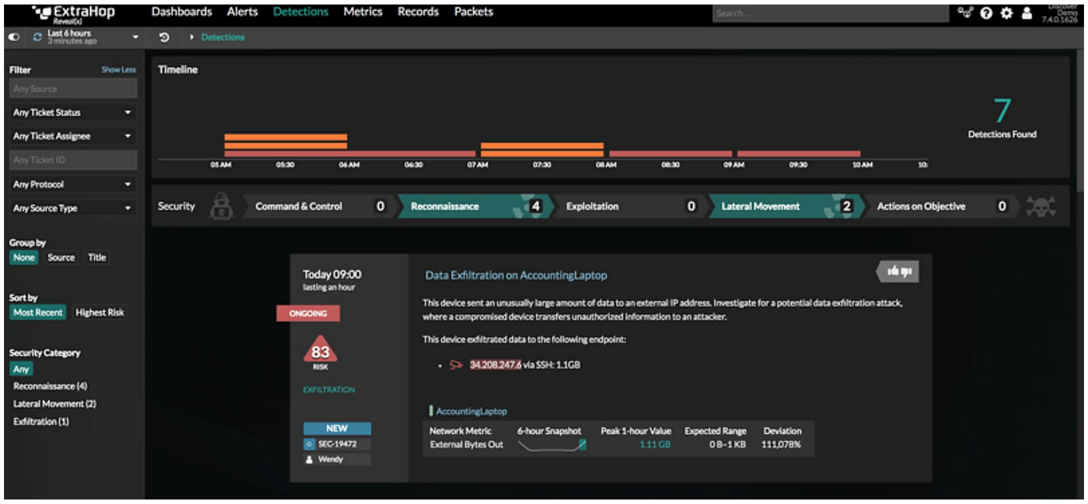
We also won't cover Reveal(x) integration with your organization's existing security tools and processes. ExtraHop has a very open integration ecosystem with partners in the SIEM, next-generation firewall (NGFW), ticketing, and orchestration and automation categories, including open access to the same API that is used by Reveal(x)'s web interface. Reveal(x)'s potential integration with your existing security tools can significantly enhance the continuity of your security operations practice and should facilitate improved automation and speed of detection to investigation.

Reveal(x) integrates with many existing SIEM, next-generation firewall, ticketing, and orchestration and automation tools, as well as supporting custom integrations through REST APIs.

Scenario 1: Breach Detection and Response

The first use case we walked through with Reveal(x) was a classic breach-detection scenario. Starting from the Event Overview dashboard, we noted a high-risk alert (a risk score of 83) related to the "AccountingLaptop" system. Reveal(x) flagged this as data exfiltration. In the same list of detections shown in Figure 2, we saw "suspicious CIFS Client File Share Access" and several "CIFS Client Access Denied Errors." We clicked the *Detections* link in the detail window to shift over to the Detections pane of the product, and then saw the detection events on a better graphical timeline (shown in Figure 5).

We were able to validate alerts by drilling into the specific details directly from the Detections list, which is packed with awesome features that help identify threats. For example, the detection for "CIFS Client Access Denied Errors" notes the error message (a Layer 7 detail) as well as information about why the behavior was flagged as anomalous. Reveal(x) conveniently breaks down the list of detections by attack lifecycle stages right at the top. In our mock test environment, our list showed Command and Control Reconnaissance, Exploitation, Lateral Movement and Actions on Objectives. A simple click on any of these categories let us directly explore the specific detection list filtered for only that category.



Data Exfiltration on AccountingLaptop

This device sent an unusually large amount of data to an external IP address. Investigate for a potential data exfiltration attack, where a compromised device transfers unauthorized information to an attacker.

This device exfiltrated data to the following endpoint:

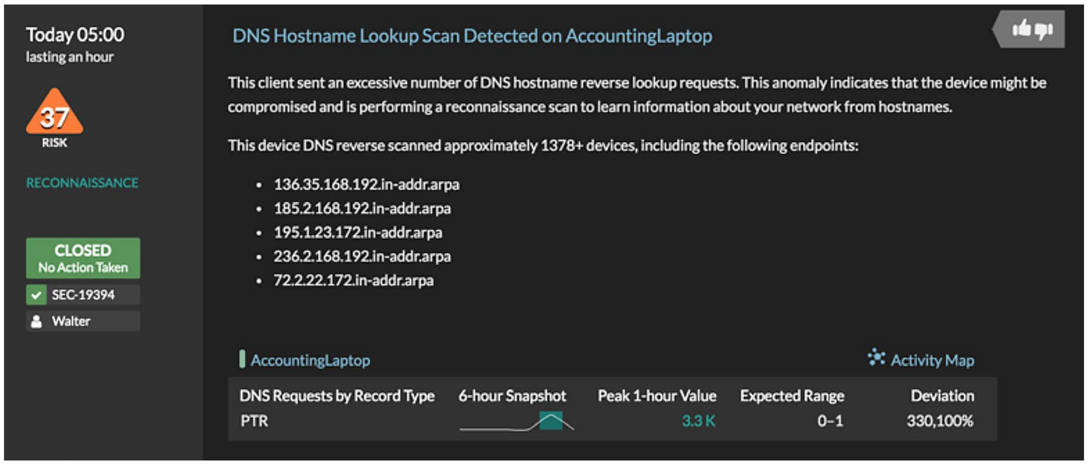
- 34.208.247.6 via SSH: 1.1GB

AccountingLaptop

Network Metric	6-hour Snapshot	Peak 1-hour Value	Expected Range	Deviation
External Bytes Out		1.11 GB	0 B–1 KB	111,078%

Figure 5. The Reveal(x) Detections Pane

We also noted that our network showed a reverse DNS lookup scan performed by a suspicious device, looking up almost 1,400 names in one hour. While this was some pretty obvious reconnaissance activity—looking for new targets or live hosts with which to coordinate command and control or exfiltration—it was encouraging to have such detailed DNS behavior readily detected by the platform, because DNS can provide a wealth of information on malicious activity going on in a busy environment. Unfortunately, DNS is often an incredibly busy protocol, with lots of traffic and logs generated, so the clues offered by DNS monitoring can often be overwhelming for security event management and monitoring tools. This is compounded by the problems of potentially not having access to logs or security teams struggling to gain access to DNS-related data in siloed team structures.



DNS Hostname Lookup Scan Detected on AccountingLaptop

This client sent an excessive number of DNS hostname reverse lookup requests. This anomaly indicates that the device might be compromised and is performing a reconnaissance scan to learn information about your network from hostnames.

This device DNS reverse scanned approximately 1378+ devices, including the following endpoints:

- 136.35.168.192.in-addr.arpa
- 185.2.168.192.in-addr.arpa
- 195.1.23.172.in-addr.arpa
- 236.2.168.192.in-addr.arpa
- 72.2.22.172.in-addr.arpa

AccountingLaptop

DNS Requests by Record Type	6-hour Snapshot	Peak 1-hour Value	Expected Range	Deviation
PTR		3.3 K	0–1	330,100%

Figure 6. Detection Card with Contextual Details

Alerting on the behavior is good, but Reveal(x) also provides contextual data such as the number of endpoints affected to help the analyst understand the scope of the event. In addition, it provides an explanation of the alert that would help a more junior analyst (see the detection card in Figure 6).

In addition, Reveal(x) has a great representation of the DNS activity for the device in a dynamic "activity map" accessible directly from the Detections pane (see in Figure 7).

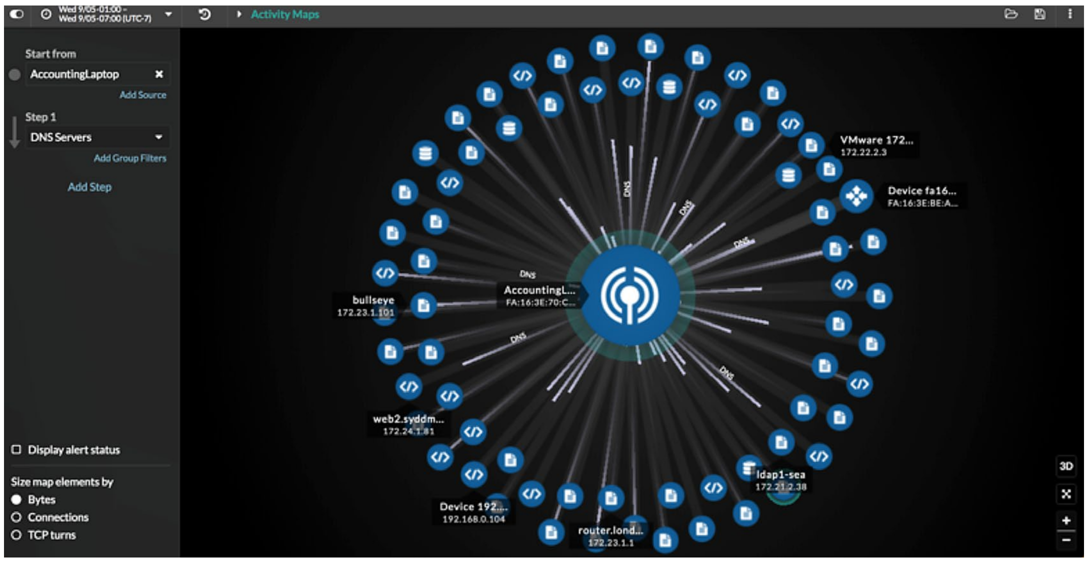
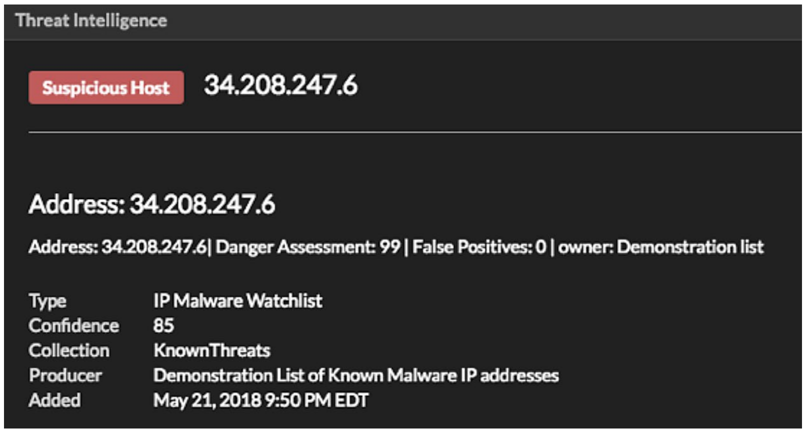


Figure 7. DNS Scanning Activity Map

Threat Intelligence

We can also trace what devices were communicating with the asset, and choose to "view threat intelligence" (the small red camera icon next to the IP address in Figure 5) related to these communicating devices based on ExtraHop intelligence and imported data from threat intel feeds that customers may have. External threat intelligence data is useful, because it can lend additional context to detected events in the environment, and Reveal(x) provides some of the correlation and telemetry that is often lacking in large, busy environments. An example of the additional threat intel context, which is accessed with a single click from the Detections pane, is shown in Figure 8.



Threat Intelligence

Suspicious Host 34.208.247.6

Address: 34.208.247.6

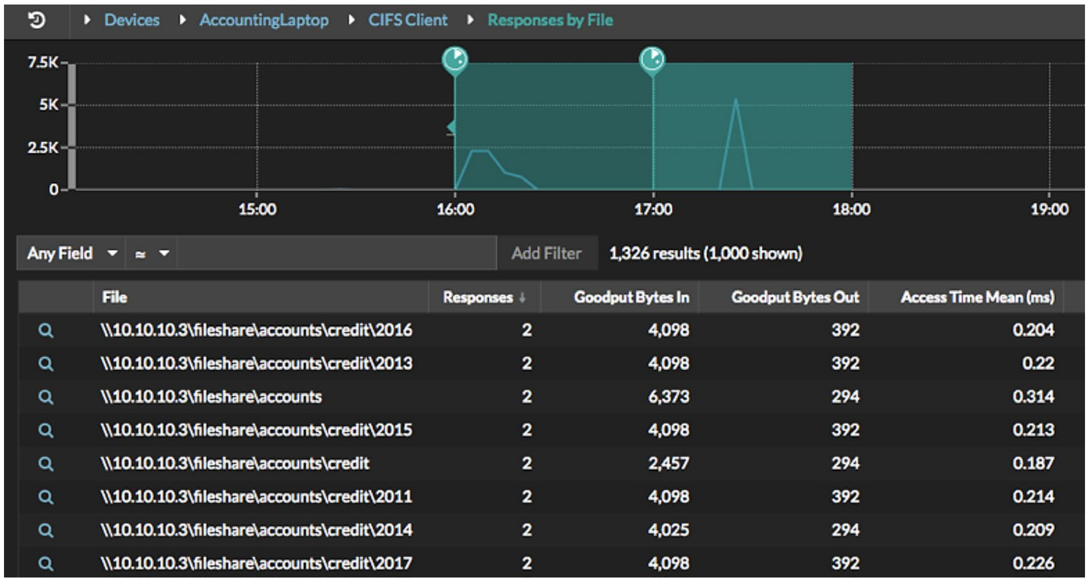
Address: 34.208.247.6| Danger Assessment: 99 | False Positives: 0 | owner: Demonstration list

Type	IP Malware Watchlist
Confidence	85
Collection	KnownThreats
Producer	Demonstration List of Known Malware IP addresses
Added	May 21, 2018 9:50 PM EDT

Figure 8. Threat Intelligence Data on a Detection

Affected Files and User Credentials

We drilled further into the incident details, where we were quickly able to discover the devices communicating and involved in the incident. We then looked at all activity for the AccountingLaptop system, and selected the CIFS activity for the specified time period. In the Metrics dashboard view, we could also select Files to see what actual files the attacker accessed, as shown in Figure 9. Layer 7 application-level information such as the filenames is extremely valuable when investigating an incident.

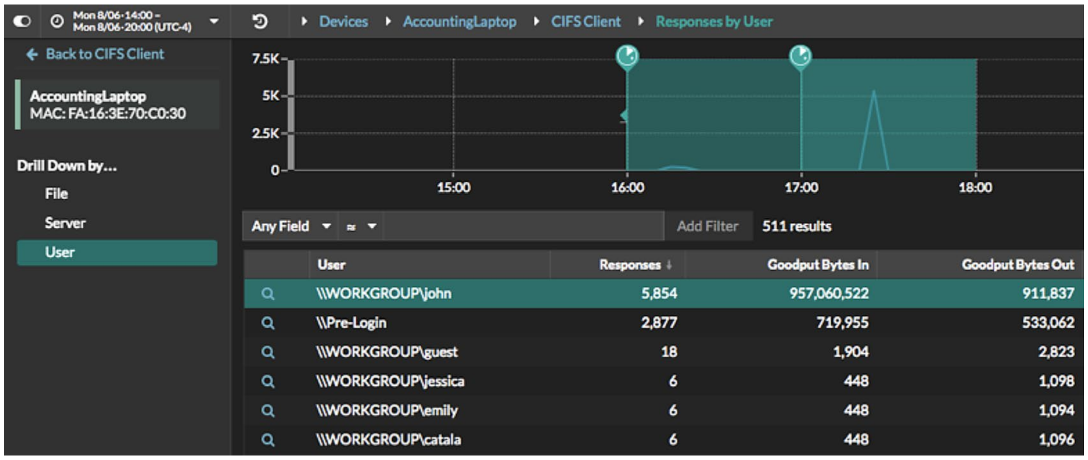


1,326 results (1,000 shown)

File	Responses	Goodput Bytes In	Goodput Bytes Out	Access Time Mean (ms)
\\10.10.10.3\fileshare\accounts\credit\2016	2	4,098	392	0.204
\\10.10.10.3\fileshare\accounts\credit\2013	2	4,098	392	0.22
\\10.10.10.3\fileshare\accounts	2	6,373	294	0.314
\\10.10.10.3\fileshare\accounts\credit\2015	2	4,098	392	0.213
\\10.10.10.3\fileshare\accounts\credit	2	2,457	294	0.187
\\10.10.10.3\fileshare\accounts\credit\2011	2	4,098	392	0.214
\\10.10.10.3\fileshare\accounts\credit\2014	2	4,025	294	0.209
\\10.10.10.3\fileshare\accounts\credit\2017	2	4,098	392	0.226

Figure 9. Files Accessed During the Breach

We then looked at the credentials used to access the file share. Reveal(x) breaks the data down by quantity of data transferred by each user credential as well (see Figure 10).



	User	Responses	Goodput Bytes In	Goodput Bytes Out
	\\WORKGROUP\john	5,854	957,060,522	911,837
	\\Pre-Login	2,877	719,955	533,062
	\\WORKGROUP\guest	18	1,904	2,823
	\\WORKGROUP\jessica	6	448	1,098
	\\WORKGROUP\emily	6	448	1,094
	\\WORKGROUP\catala	6	448	1,096

Figure 10. User Credentials Accessing File Share

Within several minutes, we were able to easily see unusual patterns of activity in the environment, find relevant contextual information (such as DNS lookups from the suspect system, threat intelligence that could lend additional insight to our investigation) and uncover critical evidence (such as systems communicating with the suspect device, files accessed and user accounts likely involved). The entire investigation was intuitive and easy to perform, and the platform demonstrated the wealth of information available to an analyst in just a small number of steps.

Scenario 2: Proactive Threat Hunting—Insider Threat

We next looked to explore a different threat scenario, where we look for insider threat behaviors that may go unnoticed by most traditional security event management and monitoring tools. Reveal(x) gathers thorough, indexed records of transactions occurring within the network environment, building on the more lightweight metrics we spent time with in our first scenario. This information provides a much more detailed forensic view of network activity that investigators can leverage in threat hunting investigations.

From the Metrics pane, we clicked the *Records* link in the upper right on our AccountingLaptop view, which then shifted our view to the Records pane itself. From here, we looked at the Filtering capability, which allows investigators to easily compose and execute queries using what ExtraHop calls a "visual query language." Analysts can save more complex searches to integrate into threat hunting workflows (see Figure 11).

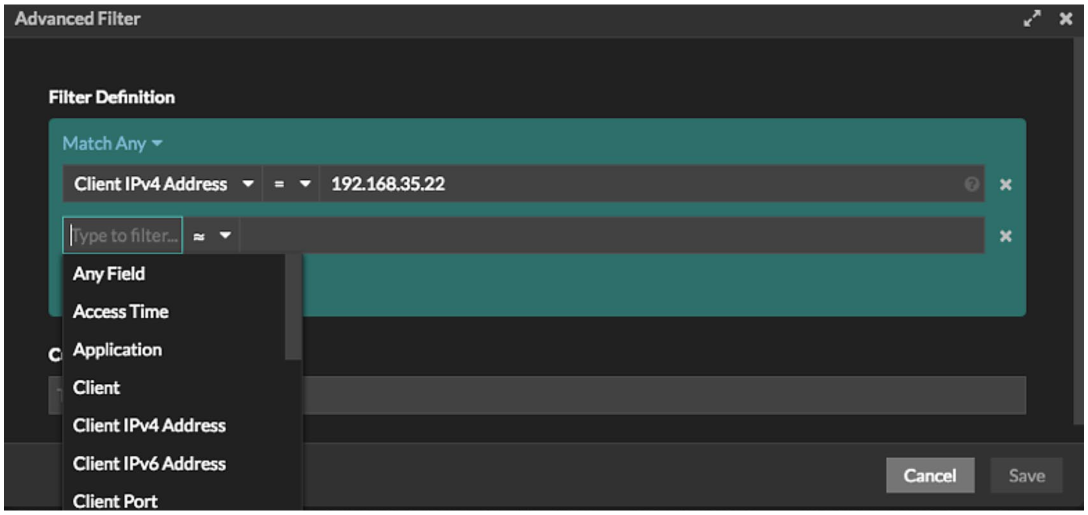
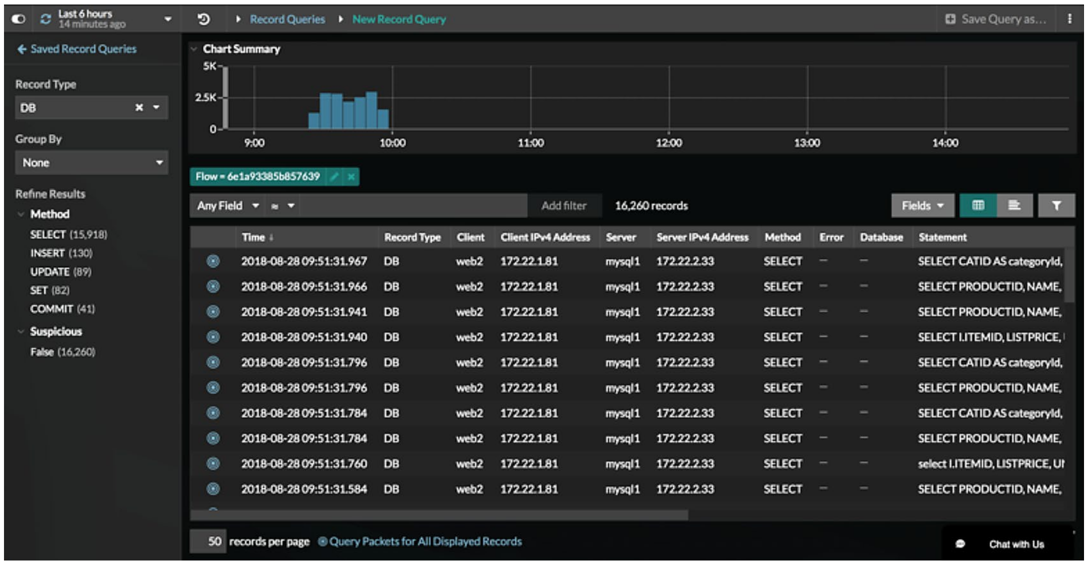


Figure 11. Creating Query Filters for Threat Hunting Investigations

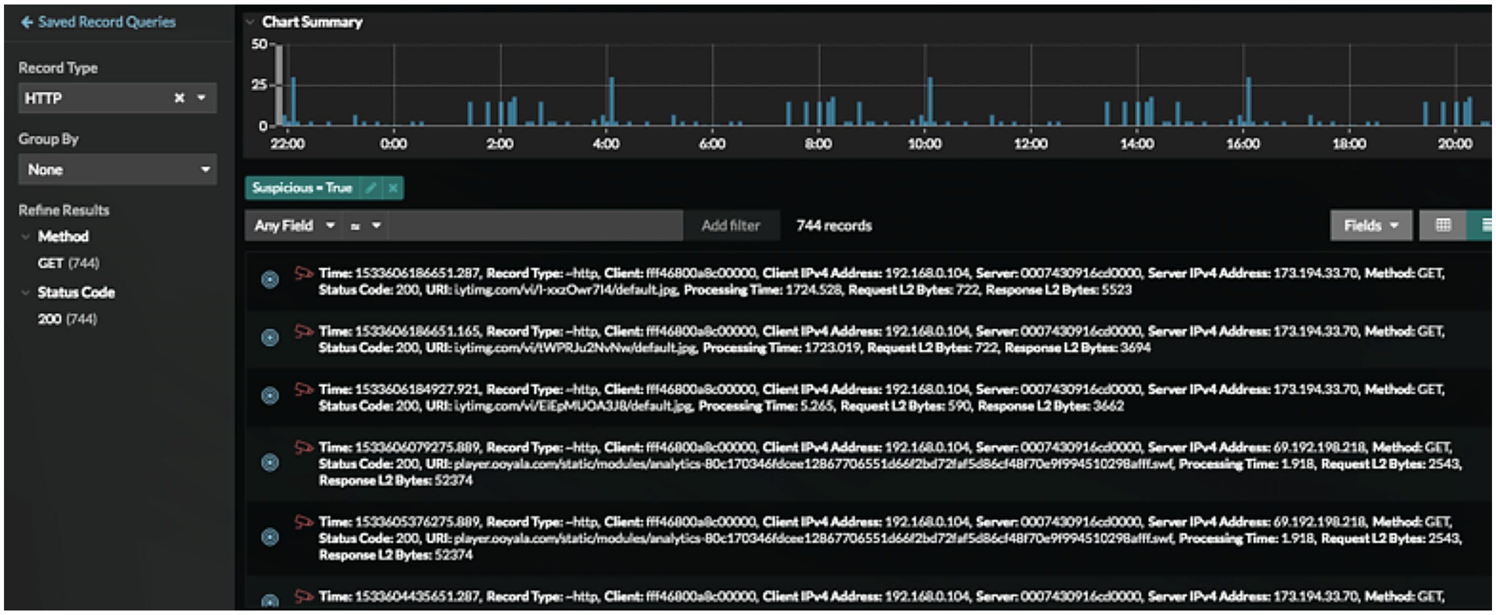
After exploring the Records view involving the AccountingLaptop system, we switched to a more generic view of all records captured by Reveal(x), and chose to look at only those associated with database activities. Once these were listed, we could choose a specific "flow" (or single session to and from a database) as shown in Figure 12.



Time	Record Type	Client	Client IPv4 Address	Server	Server IPv4 Address	Method	Error	Database	Statement
2018-08-28 09:51:31.967	DB	web2	172.22.1.81	mysql1	172.22.2.33	SELECT	–	–	SELECT CATID AS categoryId,
2018-08-28 09:51:31.966	DB	web2	172.22.1.81	mysql1	172.22.2.33	SELECT	–	–	SELECT PRODUCTID, NAME,
2018-08-28 09:51:31.941	DB	web2	172.22.1.81	mysql1	172.22.2.33	SELECT	–	–	SELECT PRODUCTID, NAME,
2018-08-28 09:51:31.940	DB	web2	172.22.1.81	mysql1	172.22.2.33	SELECT	–	–	SELECT I.ITEMID, LISTPRICE,
2018-08-28 09:51:31.796	DB	web2	172.22.1.81	mysql1	172.22.2.33	SELECT	–	–	SELECT CATID AS categoryId,
2018-08-28 09:51:31.796	DB	web2	172.22.1.81	mysql1	172.22.2.33	SELECT	–	–	SELECT PRODUCTID, NAME,
2018-08-28 09:51:31.784	DB	web2	172.22.1.81	mysql1	172.22.2.33	SELECT	–	–	SELECT CATID AS categoryId,
2018-08-28 09:51:31.784	DB	web2	172.22.1.81	mysql1	172.22.2.33	SELECT	–	–	SELECT PRODUCTID, NAME,
2018-08-28 09:51:31.760	DB	web2	172.22.1.81	mysql1	172.22.2.33	SELECT	–	–	select I.ITEMID, LISTPRICE, UI
2018-08-28 09:51:31.584	DB	web2	172.22.1.81	mysql1	172.22.2.33	SELECT	–	–	SELECT PRODUCTID, NAME,

Figure 12. Filtering Database Records for a Specific Flow

This option could easily allow an investigator to build a standard query to look for large database transactions or the quantity/volume of records pulled in a session. ExtraHop also makes it easy to query for records of any type that are flagged as suspicious. This kind of monitoring is often as difficult as some of the classic east-west and lateral movement challenges we discussed earlier, due to either the lack of access to the network activity itself or the lack of continuous monitoring within the environment. In this case, we selected HTTP records and then chose a filter to look for only suspicious records that had been flagged due to threat intelligence correlation. We did this directly in the Records pane and could view a succinct list of potential follow-ups that could be part of a threat hunting workflow (see Figure 13).



```
Time: 1533606186651.287, Record Type: ~http, Client: fff46800a8c00000, Client IPv4 Address: 192.168.0.104, Server: 0007430916cd0000, Server IPv4 Address: 173.194.33.70, Method: GET,
Status Code: 200, URI: i.ytimg.com/vi/l-xxzOwr7I4/default.jpg, Processing Time: 1724.528, Request L2 Bytes: 722, Response L2 Bytes: 5523

Time: 1533606186651.165, Record Type: ~http, Client: fff46800a8c00000, Client IPv4 Address: 192.168.0.104, Server: 0007430916cd0000, Server IPv4 Address: 173.194.33.70, Method: GET,
Status Code: 200, URI: i.ytimg.com/vi/tWPRJu2NvNw/default.jpg, Processing Time: 1723.019, Request L2 Bytes: 722, Response L2 Bytes: 3694

Time: 1533606184927.921, Record Type: ~http, Client: fff46800a8c00000, Client IPv4 Address: 192.168.0.104, Server: 0007430916cd0000, Server IPv4 Address: 173.194.33.70, Method: GET,
Status Code: 200, URI: i.ytimg.com/vi/EiEpMUOA3J8/default.jpg, Processing Time: 5.265, Request L2 Bytes: 590, Response L2 Bytes: 3662

Time: 1533606079275.889, Record Type: ~http, Client: fff46800a8c00000, Client IPv4 Address: 192.168.0.104, Server: 0007430916cd0000, Server IPv4 Address: 69.192.198.218, Method: GET,
Status Code: 200, URI: player.ooyala.com/static/modules/analytics-80c170346fdcee12867706551d66f2bd72faf5d86cf48f70e9f994510298afff.swf, Processing Time: 1.918, Request L2 Bytes: 2543,
Response L2 Bytes: 52374

Time: 1533605376275.889, Record Type: ~http, Client: fff46800a8c00000, Client IPv4 Address: 192.168.0.104, Server: 0007430916cd0000, Server IPv4 Address: 69.192.198.218, Method: GET,
Status Code: 200, URI: player.ooyala.com/static/modules/analytics-80c170346fdcee12867706551d66f2bd72faf5d86cf48f70e9f994510298afff.swf, Processing Time: 1.918, Request L2 Bytes: 2543,
Response L2 Bytes: 52374

Time: 1533604435651.287, Record Type: ~http, Client: fff46800a8c00000, Client IPv4 Address: 192.168.0.104, Server: 0007430916cd0000, Server IPv4 Address: 173.194.33.70, Method: GET,
```

Figure 13. "Suspicious" HTTP Records in Reveal(x)

We found another valuable feature accessible from within this Records pane. Once we had a query populated with records, a link at the bottom of the results page labeled *Query Packets for All Displayed Records* took us directly to the Packets pane, where we could see packet details and even download a full-packet capture of this specific dataset. This is a great example of "network recording" in action, and could really help further investigations and forensics with additional packet details.

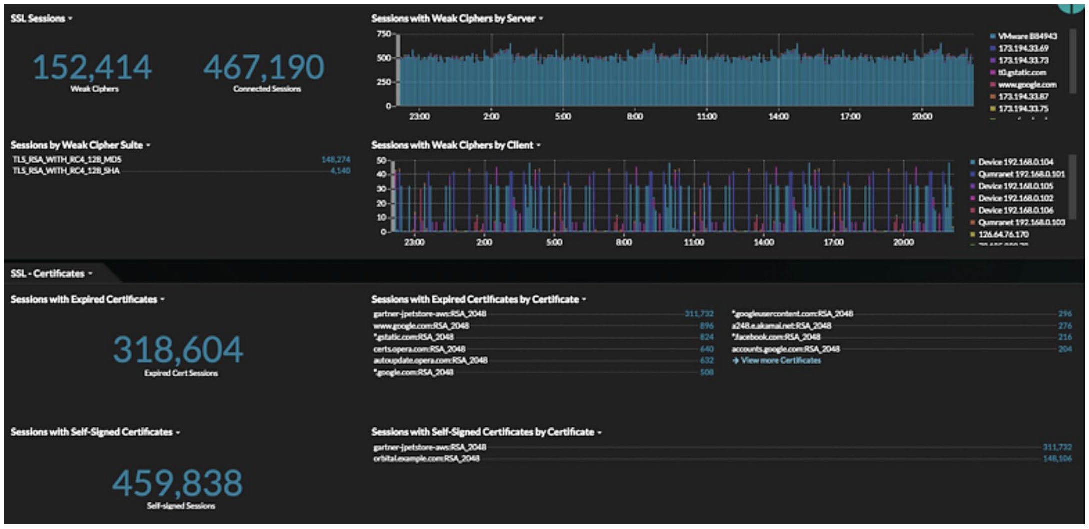
Scenario 3: Hygiene and Compliance—Reducing Attack Surface and Operational Effort

Reveal(x) has a number of additional features that could easily be used to facilitate continuous monitoring and improvement of overall cybersecurity hygiene. The 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 and best practices such as the CIS Critical Controls.

Cyber Hygiene

For example, Reveal(x) extracts detailed metadata to describe SSL/TLS certificate use, along with the 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, identify certificates that are about to expire (interrupting services), and see reports on weak ciphers being used or expired and self-signed certificates that haven't been updated or replaced (see Figure 14).

Security analysts can also leverage Reveal(x) for auto-discovery of insecure protocols being used in the environment. For example, the WannaCry ransomware that exploded in 2017 made use of the old, insecure **`SMBv1`** protocol, which Reveal(x) can easily identify and report on. For large organizations needing to find vulnerable systems using this protocol quickly, this is an invaluable feature. The same queries could also help ferret out cleartext protocols, including **`FTP`** and **`Telnet`**.

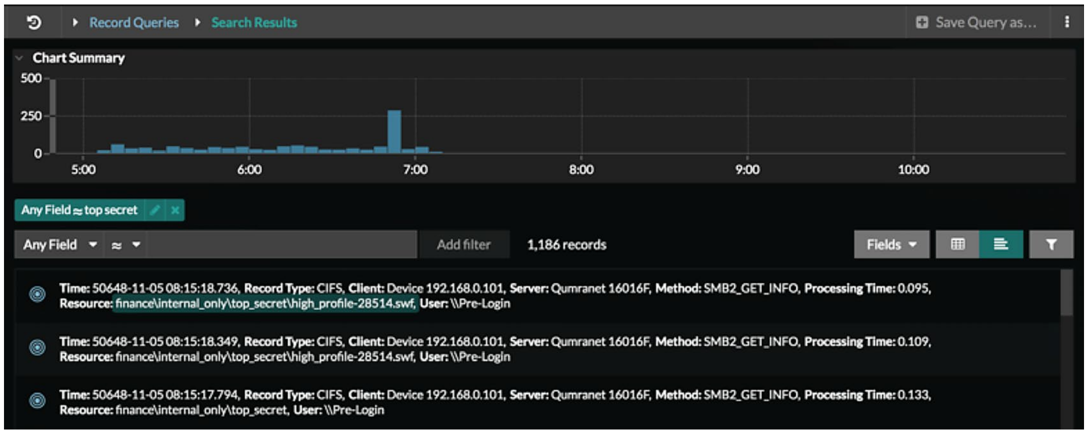


SSL Sessions
152,414
Weak Ciphers
467,190
Connected Sessions
Sessions with Weak Ciphers by Server
VMware 884943
173.194.33.69
173.194.33.73
t0.gstatic.com
www.google.com
173.194.33.87
173.194.33.75
Sessions by Weak Cipher Suite
TLS_RSA_WITH_RC4_128_MD5 148,274
TLS_RSA_WITH_RC4_128_SHA 4,140
Sessions with Weak Ciphers by Client
Device 192.168.0.104
Qumranet 192.168.0.101
Device 192.168.0.105
Device 192.168.0.102
Device 192.168.0.106
Qumranet 192.168.0.103
126.64.76.170
SSL - Certificates
Sessions with Expired Certificates
318,604
Expired Cert Sessions
Sessions with Expired Certificates by Certificate
gartner-jpetstore-aws:RSA_2048 311,732
www.google.com:RSA_2048 896
*gstatic.com:RSA_2048 824
certs.opera.com:RSA_2048 640
autoupdate.opera.com:RSA_2048 632
*google.com:RSA_2048 508
*googleusercontent.com:RSA_2048 296
a248.e.akamai.net:RSA_2048 276
*facebook.com:RSA_2048 216
accounts.google.com:RSA_2048 204
View more Certificates
Sessions with Self-Signed Certificates
459,838
Self-signed Sessions
Sessions with Self-Signed Certificates by Certificate
gartner-jpetstore-aws:RSA_2048 311,732
orbital.example.com:RSA_2048 148,106

Figure 14. SSL/TLS Cipher and Certificate Monitoring

Compliance

Reveal(x) is also a full-spectrum auditing tool for policy compliance. You can query the platform for any strings that may be seen in network data, such as sensitive content or specific filenames. We did a search on the string **`top_secret`** in our test environment and got results that show the query in file access (see Figure 15).



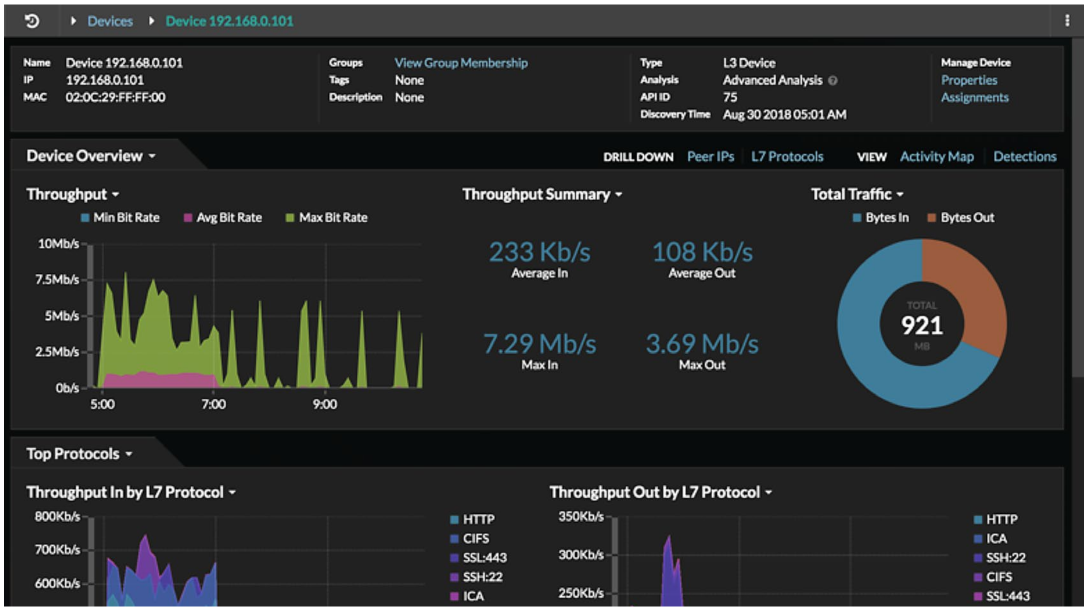
```
Time: 50648-11-05 08:15:18.736, Record Type: CIFS, Client: Device 192.168.0.101, Server: Qumranet 16016F, Method: SMB2_GET_INFO, Processing Time: 0.095,
Resource: finance\internal_only\top_secret\high_profile-28514.swf, User: \\Pre-Login

Time: 50648-11-05 08:15:18.349, Record Type: CIFS, Client: Device 192.168.0.101, Server: Qumranet 16016F, Method: SMB2_GET_INFO, Processing Time: 0.109,
Resource: finance\internal_only\top_secret\high_profile-28514.swf, User: \\Pre-Login

Time: 50648-11-05 08:15:17.794, Record Type: CIFS, Client: Device 192.168.0.101, Server: Qumranet 16016F, Method: SMB2_GET_INFO, Processing Time: 0.133,
Resource: finance\internal_only\top_secret, User: \\Pre-Login
```

Figure 15. Keyword Searches with Reveal(x)

These records also show us who accessed the file, when the file was accessed, and how many times the file was accessed—data that's not available from ordinary flow records. We can then pivot from here to see what client IP addresses were used in the file access operations and follow up on any other network activity those clients may have been engaged in. Simply by filtering on the client, then drilling into the client activity, we're brought back to the Metrics view, which shows all network traffic transmitted by the suspect client (see Figure 16).


Devices ▸ Device 192.168.0.101

Name: Device 192.168.0.101
IP: 192.168.0.101
MAC: 02:0C:29:FF:FF:00
Groups: View Group Membership
Tags: None
Description: None
Type: L3 Device
Analysis: Advanced Analysis
API ID: 75
Discovery Time: Aug 30 2018 05:01 AM

Throughput Summary
233 Kb/s Average In
108 Kb/s Average Out
7.29 Mb/s Max In
3.69 Mb/s Max Out

Total Traffic: TOTAL 921 MB

Figure 16. Pivoting on Client Filters

What this last scenario really highlighted for us, more than anything else, was the flexibility of the Reveal(x) platform. To illustrate how else the ability to query Layer 7 application-level content could be used, an analyst could compose and save a query to find any HTTP POST transactions containing **`java.lang.Runtime@getRuntime().exec`**, which is a string used in a recently disclosed Apache Struts 2 exploit. If network activity is involved, Reveal(x) collects information about it, and analysts can easily query all this information and pivot on it for security investigations.

Conclusion

Organizations are struggling to baseline behavior in their network environments so they can gain visibility and context. In particular, visibility into east-west traffic is a challenge for organizations because of encryption, traffic speed and insufficient telemetry, resulting in delays in detecting and responding to active threats. Compounding this problem is the shortage of staff and the need to simplify security operations investigations across all analyst skill levels.

ExtraHop Reveal(x) successfully addresses these issues. The tool was fast, amazingly thorough, and provided an enormous range of options for searching and querying activity within the environment. On top of that, the interface was one of the more intuitive we've used in years. We truly believe anyone could set the platform up, allow Reveal(x) to see what's going on, and start improving the security posture of their environment in no time at all.

We went into this review with the core idea of looking into lateral movement and the vexing security issue of detecting and investigating threats in east-west traffic. It’s a huge problem, and one that the security and operations communities have struggled with for years. Reveal(x) does this well, and so much more: It also helps build an asset inventory, along with protocols and applications in use. If meeting the core tenets of CIS Critical Controls 1 and 2 (and others) is a goal for your security team, Reveal(x) provides some powerful insights that make this easier.

We found the Reveal(x) platform to be detailed, flexible and helpful for any range of security operations teams who need better visibility into network behavior in their environment, with 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:

ExtraHop