WHITE PAPER

EXTRAHOP CYBER SECURITY VIGILANCE AND EARLY DETECTION EFFECTIVENESS

AN ASSESSMENT OF REVEAL(X)

JASON MACALLISTER
BEN CLEVERDON

ExtraHop

COALFIRE

North America | Europe
877.224.8077 | info@coalfire.com | coalfire.com

TABLE OF CONTENTS

EXECUTIVE SUMMARY

The landscape of cybersecurity has evolved over the last few decades. Adoption of cloud and digital services means businesses rely more than ever on network computing systems to compete at the speed of today's markets and to meet the growing demands of their customers. Likewise, public sector entities increasingly look to information technology to increase efficiency for delivery of services to their constituents. These IT modernizations further complicate security operations by introducing visibility gaps and new information silos that make the most important datasets more opaque and difficult to work with.

At the same time, for cybercriminals, business is good. The value of ill-gotten data on the black market has not decreased, and the costs to obtain valuable data have not outweighed the benefits. In fact, previously specialized toolkits are now available on the black market, changing the concern from a rare advanced persistent threat to commodity phishing, ransomware, and industry-specific targeted attacks.

In this evolving landscape, businesses are continually challenged to reduce risk to their assets without negatively impacting the value those assets provide. To do this, businesses evaluate risk by identifying threats and vulnerabilities. The viable strategy for mitigating or reducing risk takes into consideration the cost of implementing safeguards relative to the likelihood of risk occurrence and weighed against the protected asset's quantitative and qualitative value. With scant security resources, this evaluation of asset criticality is even more important. By directing more monitoring and investigation resources to the most critical assets, an organization can better balance risk and expense. While in the past companies could focus most on prevention, increased investment in detection and response supports the reality that a breach is typically thought of in terms of "not if, but when" it will occur.

One obstacle in successfully detecting an attack is visibility to the activities that occur on a network. Changing enterprise infrastructure has created blind spots, such as cloud services, micro-services, containerized apps, mobile users and devices, internet-of-things (IoT) sensors, and encrypted traffic. Because of the siloed nature of IT operations, few security teams have truly comprehensive visibility into the interactions between users, devices, applications, and the databases they use. Beyond general visibility is the need to obtain useful insights to the activities that are occurring to correctly discern malicious activity from valid activity. Many traditional detective security approaches may be agent based, use an ever-growing catalog of threat signatures, and/or employ static logic for threat alerting. It can be problematic for security personnel to wade through the cacophony of notifications. The amount of time to follow up and investigate every alert notification can be prohibitive and create alert fatigue, which can result in notifications going ignored or detection sensitivity being lowered.

To overcome these challenges, modern security technologies are being developed using artificial intelligence (AI) and machine learning capabilities to analyze network activity and automatically gain improved comprehension and insight from the abundant data available. Insights provided through this analysis can hasten the time it takes for security personnel to identify anomalous activity, investigate the activity, and respond, in a timely manner, to thwart the attackers.

ExtraHop Reveal(x) uses real-time stream processing to automatically discover and classify every transaction, flow, session, device, and asset in the enterprise, including data centers, cloud-hosted applications, remote branches, and IoT. Reveal(x) uses the richest data source available, the network, to provide the fast high fidelity insights about the internal, east-west, environment that are crucial to successful security operations. This paper reports on Coalfire's evaluation of the insights Reveal(x) extracts from the network and its ability to expedite security investigations. Reveal(x) provides an integrated solution to auto-discover and map everything on the network, analyze wire data from the network, identify anomalies using machine learning technology, map relationships between endpoints, and navigate directly to related packets to support forensic analysis of attack activities. Reveal(x) does not rely on agents; rather, it collects

an entire session of wire (L2-7) data from the network as a single correlated event using an out-of-band architecture that can decrypt traffic for immediate analysis. With coverage including cloud services, data centers, and more than 40 enterprise protocols, Coalfire found Reveal(x) integrates unique East-West visibility with intensive analysis in a model well-suited to maturing security programs.

OBJECTIVES OF THIS COALFIRE EXTRAHOP ASSESSMENT

ExtraHop Networks, Inc. (ExtraHop) engaged Coalfire Systems, Inc. (Coalfire), a respected cyber security consulting and advisory firm, to conduct an independent technical assessment of Reveal(x) with specific consideration for capabilities of Reveal(x) to analyze wire data and detect anomalies indicative of an attack. For this assessment, Coalfire reviewed documentation describing Reveal(x) capabilities, reviewed customer use cases, interviewed ExtraHop subject matter experts, and observed the Reveal(x) capabilities demonstrated in a controlled lab environment.

COALFIRE OPINION

It is Coalfire's opinion that ExtraHop's Reveal(x) can be a powerfully insightful tool for generating actionable intelligence for supporting security operations. The functionality provided by Reveal(x) can provide improved visibility for every activity that occurs on network. Moreover, the behavioral analytics capabilities of Reveal(x) can provide security intelligence to allow security operations and security analysts to focus on kill chain activities that require greater attention allowing for more timely response. Additionally, Reveal(x) discovers and increases attention on critical assets and provides situational intelligence to prioritize investigations of threats that present the greatest risk to the business, allowing security personnel to focus on and investigate the highest priority issues first. Finally, Reveal(x) investigation automation and integration support for third-party security information and event management (SIEM), orchestration, firewall, and IT service management (ITSM) solutions can support automated response work-flows to quarantine compromised systems.

No one product is capable of fully addressing all security and compliance requirements. Security is a design principle that must be addressed through carefully planned and implemented strategies. Entities seeking a strong security posture or compliance are best able to obtain it through a governance, risk, and compliance (GRC) program. For this reason, the introduction of new technologies in an organization should include the organization's security design principles to reduce risk and maintain or improve security. The benefit in this case is the ability to apply the security design principles to increase day-to-day assurance of compliance, reduce risk, and improve security. While Coalfire disclaims the generic suitability of any product for regulatory compliance or a security program, Coalfire can confirm that through careful planning, design, and implementation, Reveal(X) can be integrated into existing security systems via REST APIs. Moreover, Reveal(x) dashboards, alerts, and triggers can be customized to support entity-specific processes and policies.

OVERVIEW OF REVEAL(X) EFFECTIVENESS IN THREAT DISCOVERY

Reveal(x) provides crucial insights and automates investigations so that security teams can focus and act immediately on priority tasks. Reveal(x) uses real-time stream processing to automatically discover and classify every transaction, flow, session, device, and asset in the enterprise, including user activity, across data centers, cloud-hosted applications, remote branches, and IoT sensors. Reveal(x) provides fast, high fidelity insights about the internal East-West network transmission data that are crucial to successful security operations.

Using the network as the data source and leveraging an optional decryption appliance to see inside encrypted traffic, Reveal(x) auto-discovers assets on the network, classifying the role of each system based on the traffic observed to that device. This observed detail allows the organization to classify everything in the enterprise efficiently and accurately to correctly identify and increase monitoring of critical assets. Classifying assets by sensitivity and criticality is a pivotal step in understanding and prioritizing organizational risk so that security monitoring and staff resources can be focused for maximum business benefit.

Stronger encryption is a good thing for enterprises and, to this end, Perfect Forward Secrecy (PFS) techniques may be mandated in TLS v1.3. PFS will strengthen the privacy of traffic by creating ephemeral keys for each session so that attackers could not decrypt traffic in the future if they somehow acquire the private SSL/TLS keys. However, out-of-band analysis of network traffic is still a requirement for some compliance frameworks, and this poses a problem for traffic analytics products that cannot obtain unique session keys. To mitigate this challenge, Reveal(x) uses a software forwarder installed on monitored servers to obtain the keys for every unique session, offering a solution for inspecting PFS-protected traffic without requiring man-in-the-middle (MITM) methods. Demonstration of the Reveal(x) method was not part of the Coalfire lab test, but organizations should be aware of the issue of PFS decryption.

Reveal(x) utilizes a cloud-based anomaly detection service that takes lightweight wire data metrics and analyzes them with a suite of machine-learning algorithms to detect anomalies in time-series data for every device, network, and application on the network. In this manner, Reveal(x) can be the eyes and ears on the network, making sense of massive amounts of data and reporting on critical elements. Through the use of both next-generation behavioral and heuristic analysis along with aggregated crowdsourced feedback from its user community, Reveal(x) is innately capable of determining between what is normal network traffic and what requires immediate attention. In other words, the technology within Reveal(x) is always learning and always improving.

Using always-on machine learning on the wire data associated with critical assets allows Reveal(x) behavioral analytics capabilities to surface real threats and suspicious patterns with higher fidelity. ExtraHop account executives reflected that many Reveal(x) prospects who implement a proof of concept are surprised by the activities uncovered within minutes of collecting data. The initial insights that are discovered often reveal violations of organizational policy with identification of unauthorized traffic on network segments. In just a short amount of time, security professionals can address risks and then confirm successful implementation of security design standards.

The Reveal(x) interface guides analysts to respond to prioritized threats first, instead of progressing through a queue of unprioritized alerts. The speed at which detection is possible is important as it allows security personnel to detect an attack earlier in the kill chain. The kill chain is a model that maps common attacker behaviors from reconnaissance through data exfiltration. Whatever the technology or mechanism used, most attackers will exhibit behaviors that map well to the kill chain. Reveal(x) enables much earlier detection, giving security practitioners a better chance at containment before irreparable damage can be done.

WHY STRUCTURED WIRE DATA MATTERS

Security analytics has traditionally depended on the logging of specific security-related events in order to detect issues that occur on a device. Organizations can increase the ability to detect issues by collecting event logs from a greater number of devices in the infrastructure, such as routers, firewalls, intrusion detection systems, intrusion prevention systems, switches, servers, and client devices, among other things. Often these logs are generated through embedded services, daemons, or software agents on the device. Theoretically, data collected from multiple sources have an increased potential for improved fidelity when

the information from multiple origins is corroborated. This is largely the reasoning behind the inclusion of logging requirements for most security and compliance frameworks.

The standard approach generates a mountain of log data, making it difficult for security teams to successfully pinpoint the meaningful events necessary to produce relevant intelligence and justify appropriate action. Attackers count on this preponderance of data for concealing their activities in order to maintain a longer presence on the compromised network. Log aggregation solutions and SIEM solutions help to alleviate the burden by consolidating events to provide a single pane of glass, which allows for a more strategic analysis. SIEM correlation and analytics engines can support an increased understanding of the activities that occur on the network devices and how they might relate to one another.

However, this approach, even if successfully implemented, only portrays the part of the picture that can be described with logs and that is available to SecOps. Unless a conscientious effort is made to track new infrastructure, such as IoT devices, cloud services, and data center containers, the security team has no visibility into events or compromises outside the view of logs. Furthermore, logging inherently introduces delay and manual effort in collecting and correlating the individual data points. All of these factors contribute to reactive investigations as analysts struggle to reconstruct a complete contextual picture.

In contrast, Reveal(x) uses information collected from the transmission of network packets throughout the network. This is important, because information extracted from data being transmitted over the network is more reliable and complete. An attacker can erase logs or change what they say, whereas network traffic offers an objective look at what transpired. While there is some overlap between machine data and wire data generated from devices on the network, wire data provides additional information that is not provided by machine-generated events, such as transaction details. In addition, most organizations will not turn on verbose logging on production systems such as databases because of performance considerations, but wire data will passively collect transaction analysis including which user interacted with the database, the SQL statement, the volume of the response, methods used, and error messages.

Reveal(x) transforms transmitted network packets into wire data to provide timely insights for security operations. In modern networking, packets of information are sent across the network according to a set of transport protocols. The information contained in those packets may be structured using any number of enterprise protocols and may be encrypted for security using a range of cipher suites. Wire data is the result of decoding these wire and transport protocols and decrypting encoded data to analyze the metadata and some contents of packets crossing the network. Because of how the Transmission Control Protocol (TCP) and Internet Protocols (IP) work, in order for packets to be transformed into wire data, they must be reassembled into a per-client session or full transaction stream. This enables time-stamped measurement and data extraction, with the results indexed and stored in real-time. The resulting data set is structured, correlated to show exactly what was communicated between hosts and clients at any time, with contextual data about users, applications, and protocols involved. Reveal(x) provides at least 30 days minimum lookback into the analyzed data it has captured, with additional capacity possible by attaching network storage. The raw packets are also available via a dedicated appliance, accessible with just one click, but with more limited lookback.

As described earlier, Reveal(x) uses this wire data to perform analysis using machine learning to identify trends and highlight anomalies. The structured wire data is correlated across tiers of the network to provide better visibility into activity on the network. According to a report produced by Gartner on availability and performance data management, “while log data will certainly have a role in future monitoring and analytics, it is wire data – radically rethought and used in new ways – that will prove to be the most critical source of data for availability and performance management over the next five years.” (Cappelli, Will; Bhalla, Vivek, 2016, p. 2) Extending the logic provided in the report, wire data will play a more important role for security operations, complementing their current use of log data.

While using machine-generated data (log data, API data, instrumentation data, synthetic data) to detect issues is a traditional approach, a combined approach of using both machine data and wire data can help provide a better picture of the activities that are occurring on the network to enable more focused detection of suspicious behavior. Moreover, using wire data as part of a set of detective controls can add another protective layer of depth to a defense-in-depth security strategy. Conceptually, it adds pieces to the puzzle to help bring clarity to the scope of an organization's digital activities, both expected and, perhaps more importantly, unexpected.

For example, Reveal(x) can integrate with SIEM solutions to allow organizations to combine insights from machine-generated data with insights supplied through analysis of wire data. A SIEM will be useful for correlating the combined data, for example adding in endpoint state or registry key changes to an incident timeline, to produce more accurate and actionable information. This increased certainty of compromise allows more confidence when automating response workflows. Reveal(x) can kick off an automated response and remediation effort by sending REST API calls to an ITSM platform such as ServiceNow or to a network access control device to isolate the likely compromised systems while further investigation ensues. In turn, findings from a SIEM can be used by Reveal(x) to collect the packet-level forensic evidence to complete the investigation.

REAL-TIME ANALYSIS AND DISCOVERY

Reveal(x) draws live activity maps of the network based on the traffic at any given time. This allows network and security engineers to identify network assets and inspect the flow of traffic between devices, including the directional flow. The maps show active TCP devices (endpoints) and provide insights into what application protocols the devices are using. Reveal(x) provides a full state reassembly to replay traffic on both sides. At the time of Coalfire's evaluation, the product recognized 52 network protocols natively. Reveal(x) allows for full investigation of the network communications including which headers are present and the response formatting. Figure 1 illustrates the metrics collected from the proof of concept environment. As exhibited in Figure 1, all of the TCP connections, both inbound and outbound are displayed. The column on the left allows the user to drill down into the data to see server and client activity for auto-discovered protocols (in this case CIFS, database, DNS, HTTP, LDAP, SSL, and DHCP), both for server activity and client activity.

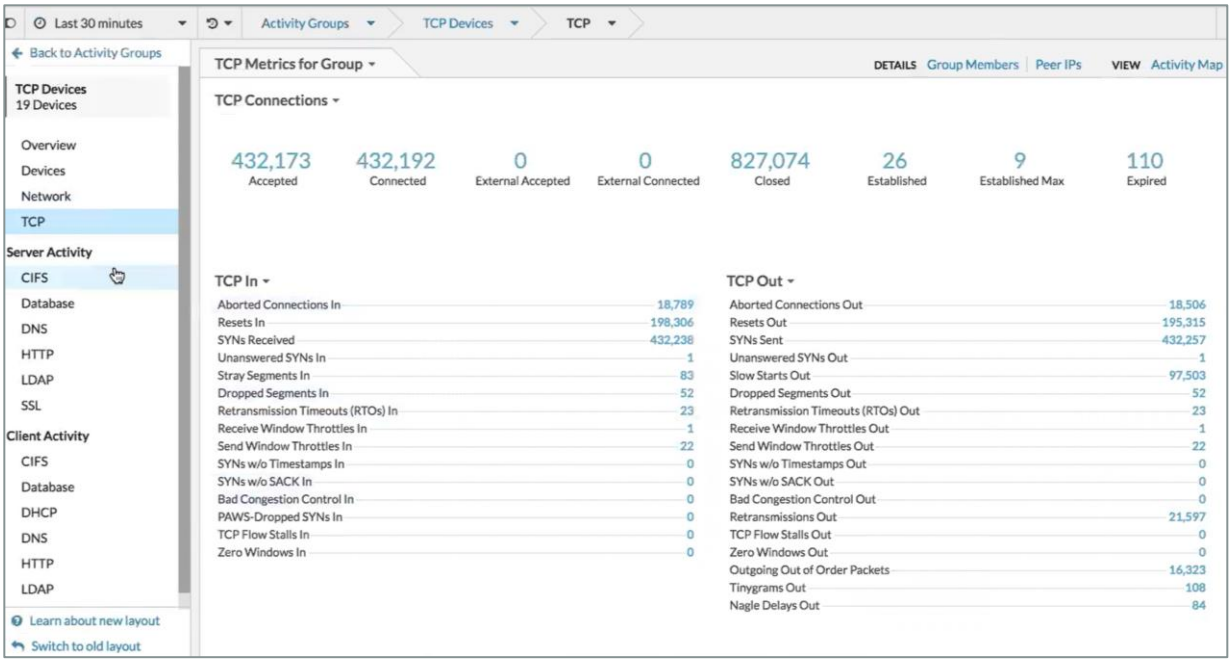

Figure 1 – Sample Reveal(x) metrics

Figure 2 provides a graphical illustration of the activity that is occurring on the network. This is an example of the live activity maps. This particular map illustrates the three distinct environments that make up the proof of concept environment. The circles in the map represent devices (servers and clients) on the network. The lines between the circles show the flows of network traffic between these devices. Looking closer at the lines provides specific information about the type and number of transactions that are happening between the endpoints. This map is interactive in that it allows for the user to view specific details about that device and its connections.

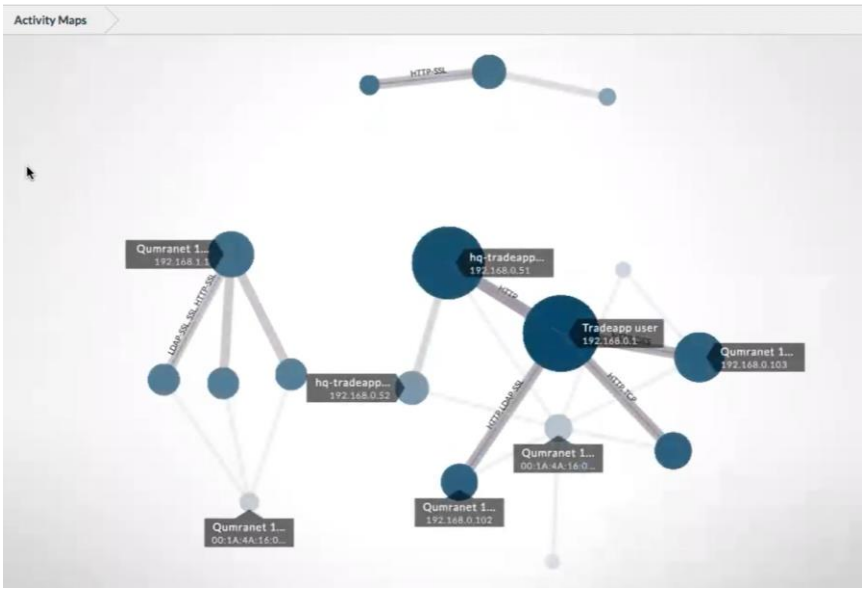

Figure 2 - Activity maps graphically Illustrate what is happening on the network

Reveal(x) can understand context for the observed traffic to determine whether the source of network communication is a server or client, as well as its role, such as a file server or an application server. This is illustrated in Figure 2 above. This is important for Reveal(x) to determine expected and unexpected behaviors. For instance, it would be unusual for a server to be running a client process, so this may be reported as an anomaly requiring additional investigation.

Once a suspicious or anomalous activity is discovered, Reveal(x) allows for a full replay of activity including the client, server, and context of the communication from the console. This is useful for further investigation of discovered anomalies to determine exactly who, what, when, and how something happened on the network. For example: a discovered HTTP client connection was made from a server object to a foreign URL. This was tagged by Reveal(x) as an anomaly because a server should not be acting as a client. The replay of the session revealed specifics about the URL, the request that was made on the target, and the nature of the package that was downloaded to the server. Investigation of ensuing events revealed additional steps taken by the attacker following the kill chain.

TESTING SCOPE AND METHODOLOGY

Coalfire observed a Reveal(x) proof-of-concept deployment for one of its clients. The scope of Coalfire's review included the elements of Reveal(x) that were current at the time of this publication. Coalfire's review primarily focused on the capabilities of the product to detect anomalies that are indicative of the patterns of attack in real world scenarios for improving visibility for security incidents, which could be useful for decreasing time to containment, and support post-containment deep invesigation.

CHARACTERIZATION OF TEST LAB

The test lab was made up of a small data center infrastructure with various devices on the network representing common business applications including multi-tiered web applications, e-commerce services, CIFS servers, DNS servers, directory services servers, and client devices, among other relevant components. Also placed throughout the lab environment were the various ExtraHop appliances necesssary for supporting network discovery, visibility, insights and review of collected wire data, and connectivity to the cloud-based anomaly detection engine for analysis. The lab included three DMZs. One DMZ was placed in an Amazon Web Services (AWS) virtual private cloud (VPC), one DMZ in a Seattle, WA data center, and a DMZ placed in a simulated branch office located in Sydney, Australia. The architecture was intended to represent a hybrid implementation of a public cloud, a private on-premises infrastructure, and a remote or branch office. This is largely notional, but can be satisfactorily extended to represent a wider variety of IT structures. Each location included various workloads.

To generate "normal" traffic on the network, bots were created to simulate a day-to-day operation for the business. The bots carried out various routines against the network such as browsing web applications, logging onto web applications, navigating the application, placing orders on the e-commerce applications, and interacting with files on CIFS servers. The activity of the bots was randomized enough to optimally simulate real-world traffic. Once the environment was set up and normal user activity was generated, a period of time was allowed to elapse wherein Reveal(x) collected wire data. Also during this time, Reveal(x) was analyzing the presented wire data over prescribed intervals. This allowed the product to identify normal behavior patterns and establish a baseline for comparison.

There was also a subset of scripts that were run from clients on the network to represent attackers on the network who would carry out kill chain tasks representing an advanced persistent threat scenario. Reveal(x) was primarily collecting data representative of East-West traffic on the network and this was the target of its analysis. The scripts were set to replay security issues over randomized intervals – for example, at least once every 12 hours offset by 10 hours.

VALIDATION EXERCISE FINDINGS

The following narrative and screenshots depict the findings from the proof of concept with respect to security incident detection. The screenshots are taken from the Reveal(x) dashboards and user interfaces. The information tells a story of a point in time on the network when an attacker set out to breach its intended target. The story highlights the machine learning functionality of Reveal(x) to detect and report on anomalies, which allows security personnel to focus efforts on specific network events that have a higher probability of indicating an in progress attack.

Figure 3 is taken from the Reveal(x) user interface. This figure depicts the detection of anomalous activities and was presented to the organization in the form of an alert. Reveal(x) was able to determine, against baseline activity, that this was unusual activity that was occuring on the network over an extended time period. Unlike hard-to-interpret logs and packets, the UI presents the information in terms a Tier 1 analyst can easily evaluate, including possible security implications of this specific type of anomaly and the degree to which the activity was abnormal. A high deviation affecting critical assets generates high confidence that a situation is worth investigating.

This example details a spike in SSH server sessions on a number of servers in the network. The screenshot shows the duration of the activity, the destination or targeted device for the activity, a summary finding for the anomaly, and some details (protocol and client). The client in this case is likely the machine from which the attacker has established a foothold. With an identified anomaly, further investigation begins to provide more detail indicating what happened.

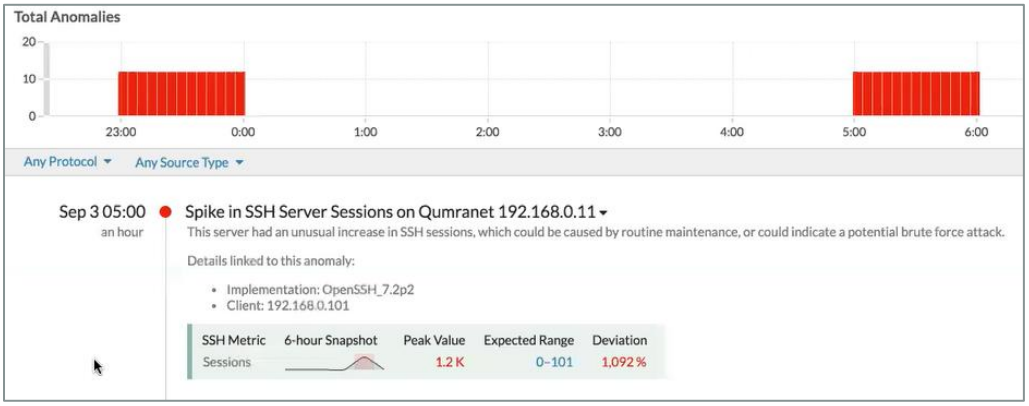

Figure 3 - Reveal(x) interface graphically illustrating anomalies

Looking at the source of the attempted SSH connections provides a picture showing that a server in the environment is acting as an SSH client. It is reasonable to assume that a server should not act as a client and therefore this alone is abnormal behavior. Reveal(x) alerts that the server has an unusual increase in SSH sessions that may be indicative of a potential brute force attack. The fact that there was not any activity previously of this nature and suddenly a spike in activity occurs further supports concern.

The attacker in this case has likely enumerated the devices on the network and chosen to run SSH connections against the servers to find a device for which he or she can establish a connection. Figure 4 depicts the device where the attempted SSH sessions are originating. It also shows all the servers that are linked to the anomaly. These would be the targeted servers.

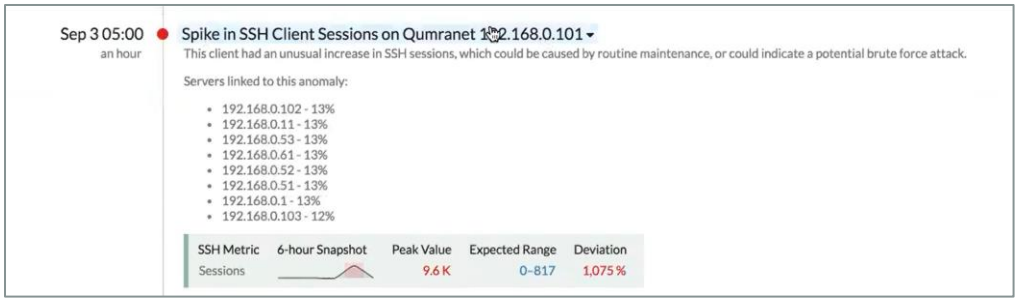

Figure 4 - Depiction of all the SSH connections from an SSH client on a server

Inspecting the information on the specific server where the SSH sessions originate allows the Reveal(x) user to investigate further and identify other activities that are occurring on this server. The user can pull up metrics for the server from Reveal(x) as shown in Figure 5. An awareness of the number of SSH sessions from this device is confirmed through the metrics view with over 19,193 sessions over the time period with unusual spikes in activity over a specific interval. Additional metrics can be looked at in this view for other observed traffic that is related to this server to assess whether other unusual activity is occurring.

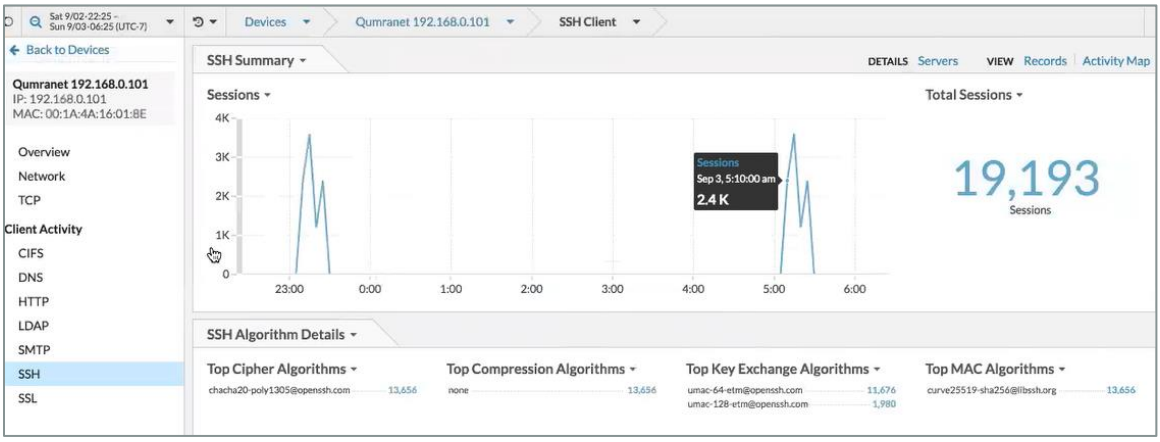

Figure 5 - Metrics detail for the server with suspicious activity

Optionally, an activity map can be pulled up to show the offending device on the network and all the connections that it is making. Figure 6 shows the attempted connections out to the other devices on the network using SSH. There is also another connection from the server that is in the background in gray. Further investigation can reveal what else may be happening on the server.

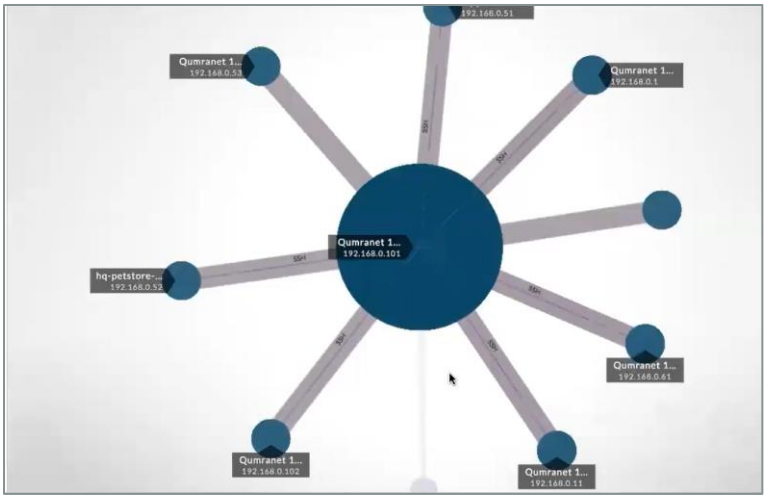

Figure 6 - Activity map with suspicious server at the center

In addition to these off the shelf views, security dashboards can be created to highlight specific information that might be of interest to the organization. This allows for a focus on key information with an at-a-glance view via charts that quickly reveal the activity of the server. These dashboards can provide visibility into critical indicators that are useful for narrowing down suspicious activity. What is already known is that a server on the network began acting as an SSH client and attempted a number of connections to other servers on the network. Examining the dashboard reveals the other suspicious activity that occurred on the network. Figure 7 depicts a dashboard view for the suspicious server. This particular image illustrates the HTTP critical indicators related to the server. Other views also depict SSH activity and SSL activity.

By investigating what is unknown and attempting to understand what else is occurring on the server, Reveal(x) has highlighted some suspicious external HTTP requests from the server. Among the suspicious requests are connections out to Russian web sites. Of particular interest is a connection to what appears to be a Russian fileshare (highlighted by the orange oval in Figure 7).

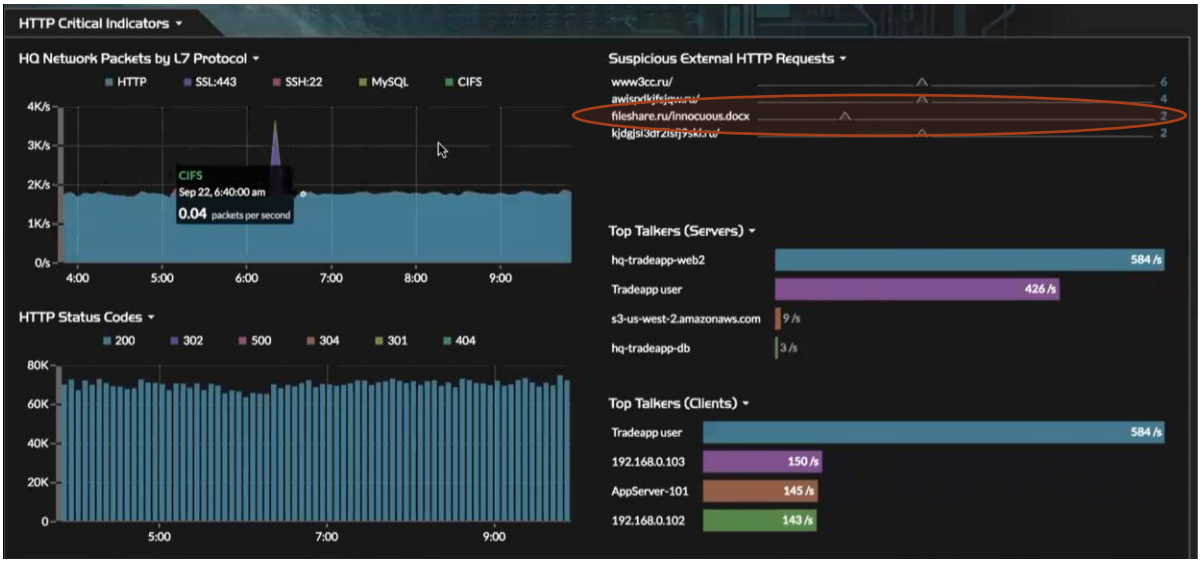

Figure 7 - Dashboard view of critical indicators

Right-clicking the particular Suspicious External HTTP Request brings up a context menu for which the Records related to the HTTP request can be pulled up as shown in Figure 8.

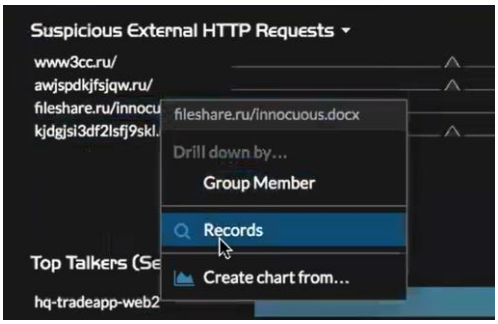

Figure 8 - Selecting the request records

This brings up all the records associated with the request. As exhibited in Figure 9, this view presents the time and date, the record type, where the request is coming from, where the request went to, the method of connection, status code, byte size, and what the file was. Network analytics products that only look at flow data would not conduct deep enough analysis to reveal the name of the file in question, which could be vital information for determining the impact of an incident.

Packets	Time	Record Type	Node	Client	Client IPv4 Address	Server	Server IPv4 Address	Method	Status Code	URI	Proce
	2017-0[illegible]22 05:07:01.693	HTTP	eda-hq	AppServer-101	192.168.0.101	Qumranet 192.168.0.61	192.168.0.61	GET	200	fileshare.ru/i...	

Figure 9 - HTTP request records

From the dashboard, the specific packets for the transaction can be grabbed from the packet capture (pcap) via a single click. In this case, the grab of packets pulled twelve packets for the specific transaction that the security administrator can look through. Reveal(x) was able to pull up the related packets in a matter of seconds. This is fairly impressive considering that the packets were queried from 5.11 GB of data representing over 19.4 million packets at the time of the lookup. Pulling up the packets allows SecOps personnel to see exactly which file was grabbed, which helps to identify with what the server was infected.

Deeper investigation into the SSH sessions shows, for all the attempted connections, one session lasted significantly longer than all the other sessions. This is indicative of a successful connection with enough time to execute a script or malicious code on the target. Figure 10 shows the session attempts. The highlighted session is the session of particular interest.

SSH Session Duration by Server

Server IP	Host	Sessions	Session Duration Mean (ms)
192.168.0.103	192.168.0.103	1,205	0.4
192.168.0.61	192.168.0.61	1,200	0.082
192.168.0.53	192.168.0.53	1,200	0.183
192.168.0.102	192.168.0.102	1,200	0.206
192.168.0.52	192.168.0.52	1,200	0.179
192.168.0.51	192.168.0.51	1,200	0.219
192.168.0.1	192.168.0.1	1,200	0.078

Figure 10 - Session attempts

Following the breadcrumbs, the investigator can now look at the activity on the device where the attacker was able to make a successful SSH connection. Figure 11 shows the metrics from Reveal(x) for this device. Looking at the group activity on the far left panel, it appears based on the client activity that this is a client device. However, it also appears that a client device is now acting as an SSH server, which denotes suspicious activity. The metrics show a sudden spike in activity that can be further investigated.

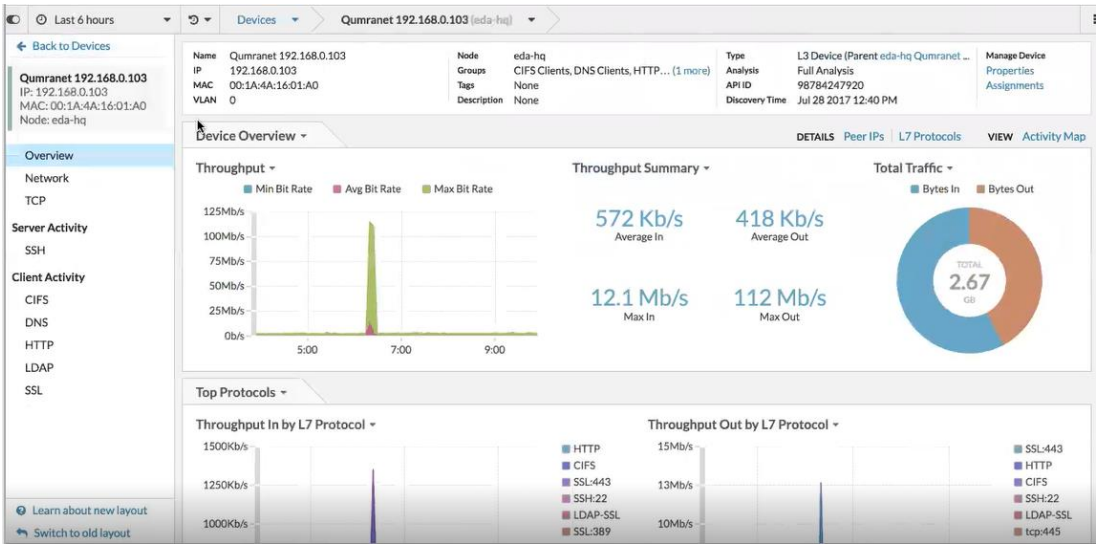

Figure 11 - Metrics for next step in the attack process

In scrutinizing the activity of the device in the metrics dashboard, there are several items of interest. Of primary interest is a sudden spike in CIFS traffic (as shown in Figure 12). This graph indicates that there was little to no CIFS traffic and then an abrupt spike in traffic.

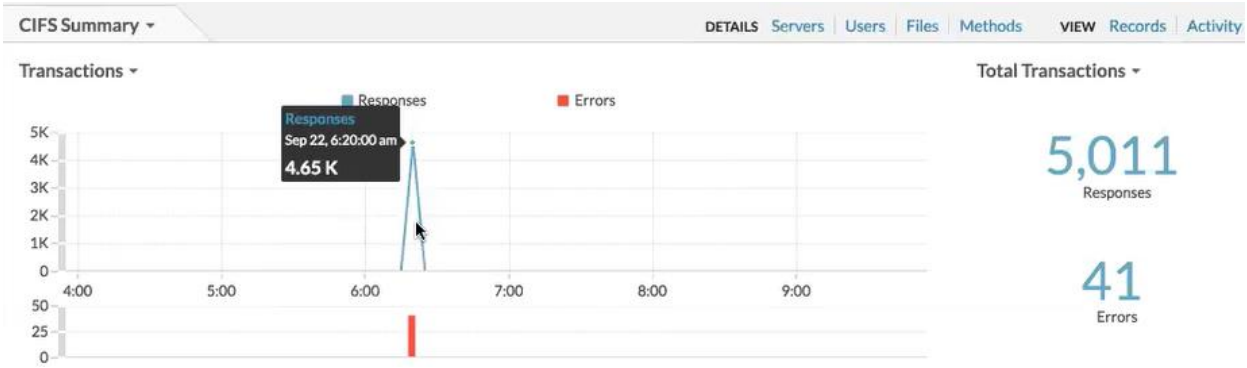

Figure 12 - CIFS metrics

Drilling down on the spike in activity reveals that the attacker enumerated the finance directory and then began to grab content from the finance directory as shown in Figure 13.

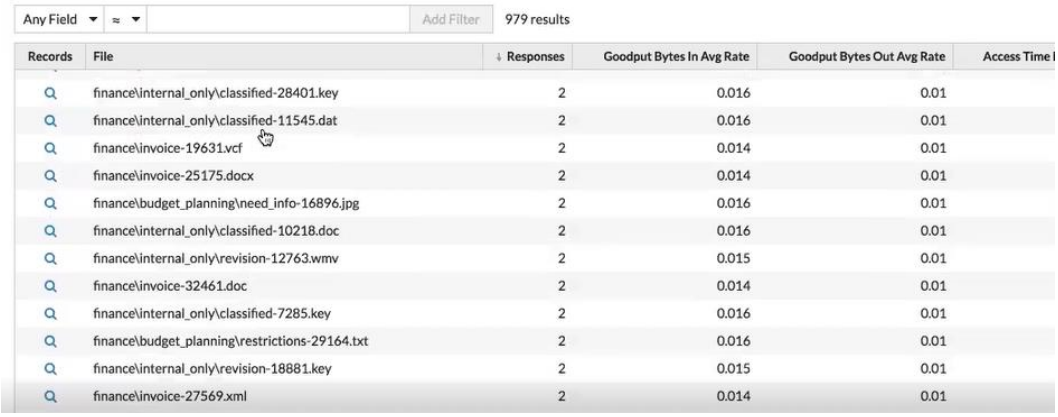

Figure 13: File activity represented by the spike in CIFS activity

Also of interest for this device was a sudden and unexpected spike in SSL traffic as shown in Figure 14. Selecting this metric provides the ability to see to where the connections were made.

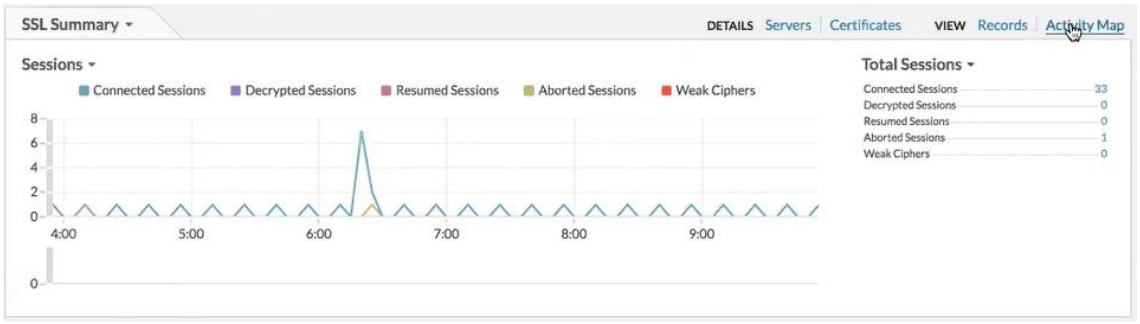

Figure 14 - SSL traffic from the same device

Figure 15 shows the connection detail in the live activity map for the SSL connection to an external target. This target represents an AWS S3 bucket.

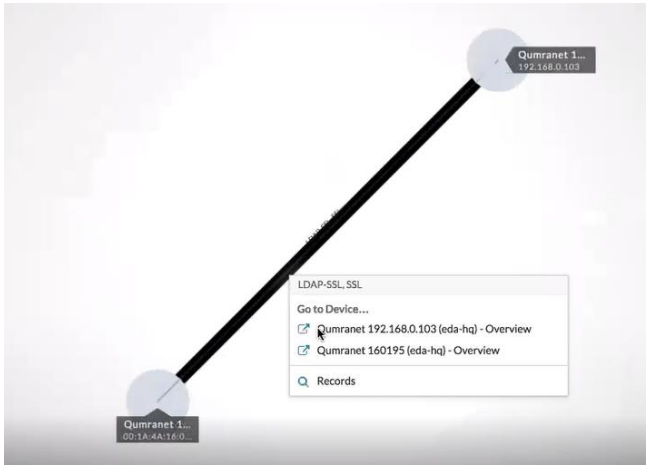

Figure 15: SSL connection detail

Reveal(x) provided the intelligence analytics to allow security personnel to focus on important information: the attacker's entrance point, the file that was used to infect the target, the device that was infected, what the infection did, how the attacker was allowed to access sensitive data, and where the attacker was able

to exfiltrate data. All of this investigation started with the identification of numerous connections that should not have ordinarily been present in the sample network.

In this case, Reveal(x) knew that certain connections should not have been occurring and reported that the anomalies should be further investigated through an alerting process. The product's capabilities to perform this analysis through machine learning reduces the necessity for human intervention through instrumentation to detect security issues, while also facilitating the investigation through live activity maps and record search.

ADDITIONAL CONSIDERATIONS

Of particular consideration is the importance of the organization to "mind the shop". Any tool is useful, but only if it is used. To use a tool, the staff assigned the responsibility for the tool require training and accountability to be successful. It is also important that organizations consider the role that a technical solution plays within the overall strategy of the organization and how it will be useful to complement the organization's existing arsenal of tools. The ability to integrate and automate processes among the existing tools, to save manual effort for understaffed teams, is critical. Reveal(x) provides a powerful and simple API to allow all of its data and insights to be accessible to automation and orchestration systems, SIEMs, and other must-have security tools.

CONCLUSION

Having a broader range of security options available that are able to work together to enable better decision making and improved response time can help minimize the impact of cyber attacks. One of the greatest obstacles to timely containment of security incidents is a lack of visibility as to what is happening in an environment. Improved visibility, such as the analysis and reporting provided by ExtraHop's Reveal(x), enables awareness of all activities that occur on the network and can help identify any security incidents as they transpire. Moreover, the machine learning capabilities of Reveal(x) help to narrow the focus for watching the network to what is statistically important.

The competitive differentiation of Reveal(x) lies at the intersection of visibility and advanced analytics. Since the product gains deeper visibility into each transaction by decoding protocols and decrypting traffic, the insights that are automatically derived from the data via machine learning are commensurately richer than using event driven logging and analytics alone. This combination of visibility, analytic power, and the ability to access all the relevant data from a single UI makes Reveal(x) a compelling centerpiece for next-generation security programs.

RESOURCES

www.extrahop.com

https://www.extrahop.com/solutions/initiative/security/

https://www.extrahop.com/platform/addy-machine-learning/

https://www.extrahop.com/company/blog/2015/the-four-data-sets-essential-for-it-operations-analytics-itoa/

https://www.gartner.com/doc/3245417

https://docs.extrahop.com/7.0/dep-eda-6100-8100-9100/

BIBLIOGRAPHY

Cappelli, Will; Bhalla, Vivek. (2016). *Use Data- and Analytics-Centric Processes With a Focus on Wire Data to Future-Proof Availability and Performance Management.* Gartner, Inc. Stamford: Gartner, Inc. G00302399.

ABOUT THE AUTHORS

Jason Macallister | Author | Senior Consultant, Cyber Engineering, Coalfire

Mr. Macallister consults on information security and regulatory compliance topics as they relate to advanced infrastructure, emerging technology, and cloud solutions.

Chris Krueger | Managing Principal | Principal, Cyber Engineering, Coalfire

As Principal, Mr. Krueger contributes as an author and thought leader on information security and regulatory compliance topics for Coalfire's clientele in the "new and emerging" technology areas.

Ben Cleverdon | Technical QA | Consultant, Cyber Engineering, Coalfire

Mr. Cleverdon consults on the security and integrity of both cloud and physical infrastructures, and assesses vulnerability management and compliance framework implementation.

Published March 2018

ABOUT COALFIRE

Coalfire is the cybersecurity advisor that helps private and public sector organizations avert threats, close gaps, and effectively manage risk. By providing independent and tailored advice, assessments, technical testing, and cyber engineering services, we help clients develop scalable programs that improve their security posture, achieve their business objectives, and fuel their continued success. Coalfire has been a cybersecurity thought leader for more than 16 years and has offices throughout the United States and Europe. Coalfire.com