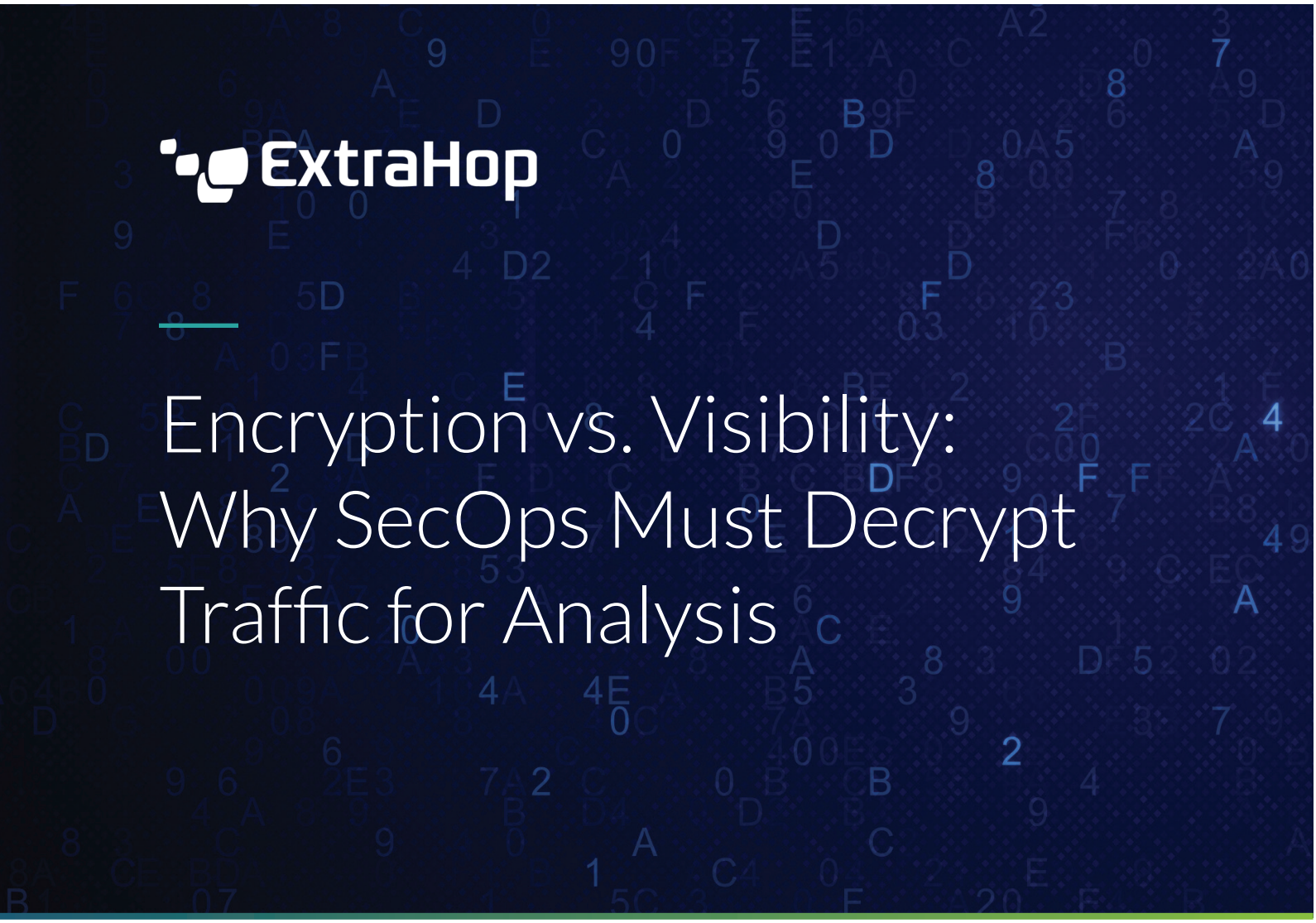


Encryption vs. Visibility: Why SecOps Must Decrypt Traffic for Analysis

ABSTRACT

Research shows that enterprises are increasingly encrypting traffic inside corporate networks (the east-west corridor), on the public internet, and in the north-south channel between them. At the same time, attackers are increasingly using encrypted traffic to hide their malicious acts.

In this paper, you'll learn about several options for retaining and expanding the needed visibility to detect and respond to threats in encrypted traffic. You'll also learn how ExtraHop Reveal(x) decrypts traffic in real time for out-of-band analysis, with no latency issues, to enable SecOps to see and fight threats that are hiding in the encrypted dark space.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Encryption is being used more frequently, both on the public internet and inside of corporate networks, driven by the common knowledge that strong encryption is a sure-fire way to help protect data and information. However, the increasing use of encryption has the secondary effect of making security monitoring more difficult. Cyber adversaries leave telltale signs in the network traffic of the enterprises they attack, and encrypted data makes it more difficult for security teams to detect these signs. This conundrum has led to a rise in cybersecurity products offering two remedies:

- Decrypt internal traffic for security analysis to detect threats
- Analyze encrypted traffic and infer security information based on analyses that can be done without decrypting the data (sometimes called encrypted traffic analysis, or ETA)

Decrypting traffic for analysis offers much richer threat detection, investigation, and response capabilities. The ETA approach, on the other hand, can uncover some threat indicators, but these are not adequate for detecting today's advanced threats. This paper gives technical details on both approaches and makes the case that decrypting internal east-west traffic offers a deeper analysis, introduces very little risk, does not introduce data privacy concerns, and is by far the superior approach.

Within the decryption approach, there is two-pronged debate about how best to implement decryption:

- Decrypt traffic in-line by terminating sessions at a firewall or proxy, then re-encrypting before sending the traffic to its final destination
- Decrypt a copy of network traffic out-of-band without impacting the flow of that traffic

Out-of-band decryption and analysis is more secure, enables richer analysis, and does not impact the performance of the network. In-line decryption frequently degrades the overall security of the data in transit, and increases network latency.

This paper will go into detail about each facet of these debates, and provide industry evidence and technical explanations and illustrations to demonstrate that Reveal(x) 360 takes the best approach to detecting threat behaviors in network data in the enterprise—and offers the best implementation of that approach.

WHY ENTERPRISES ARE RAPIDLY ENABLING STRONG ENCRYPTION

In the past, and even today, many enterprises neglected to encrypt the traffic traversing the east-west corridor inside their network. Encrypting data takes work, introduces complexity and cost, and reduces the visibility that security operations teams need when monitoring their systems and data.

However, as general concerns about data privacy grow and regulations like the General Data Protection Regulation (GDPR), the Health Insurance Portability and Accountability (HIPAA), and the Payment Card Industry Data Security Standard (PCI DSS) have come into effect , the adoption of in-flight data encryption on the web and inside the enterprise has increased. This effect is reinforced by recent updates to Microsoft protocols like SMB which reduce the complexity and management overhead of enabling encryption on organizations internal networks. It's: understandable, then, that SecOps teams may view encrypted east-west traffic as a double-edged sword: it's a necessary part of their jobs, but it also makes their jobs more difficult.

Today, the majority of web traffic is encrypted, a trend driven by major web technology providers. The Google Transparency Report says that 95% of web traffic to Google in the United States is encrypted, with similarly high percentages of encrypted requests from many other countries worldwide.

Data center traffic is also increasingly encrypted as organizations respond to regulatory and customer requirements, and more technology vendors turn encryption on by default. A 2021 Ponemon study found that the number of businesses applying encryption across their enterprise networks has increased steadily since 2005 across all industry sectors, from 15% in 2005 to 50% in 2020. A 2019 survey report issued by Enterprise Management Associates (EMA) indicated that 59% of very large enterprises already had TLS 1.3 encryption enabled, 74% of respondents had either already started enabling TLS 1.3 encryption on internal connections or were planning to within six months.

These trends are reinforced by the deprecation of TLS 1.0 and 1.1 by the IETF in RFC 8996, prompting the rapid end-of-life support for these vulnerable legacy protocols by many industry vendors including Microsoft. Furthermore, perfect forward secrecy (PFS) has been available as a feature of TLS 1.2 for years, and many enterprises have already enabled it, while TLS 1.3 has adopted PFS as a requirement. These changes, while beneficial and necessary, have created a real visibility challenge for security teams.

DARK SPACE:
WHY DECRYPTION IS NECESSARY FOR SECOPS SUCCESS

While encryption is a good thing for privacy, it's also a boon to hackers. Encryption, both inside corporate networks and on the public internet, creates dark spaces and blind spots that attackers use to hide their activities from security teams. The 2018 Annual Cybersecurity Report from Cisco showed that 70% of the malware binaries they sampled took advantage of encrypted network traffic in some manner. Zscaler reports the blocking of 733 million encrypted attacks per month in 2020, a 260% increase over 2019. Furthermore, Kaspersky reported in 2020 that 38.6% of attacks utilized "living off the land" strategies that leveraged existing systems and technology inside their target networks, which allowed them to move laterally and escalate privileges. Encryption is vital for security and privacy, but it can be a double-edged sword when attackers are able to hide their malicious actions in legitimate-seeming encrypted traffic using approved capabilities in their target networks.

Because encryption is now widely used in both data at rest (databases and storage), as well as data in transit (internal authentication activity associated with lateral movement, data staging, and application use), it is clearly vital for security teams to have visibility into encrypted traffic. Analyzing the decrypted contents of transactions across the network allows for faster identification and remediation of threats before a headline-making data breach happens.

WHAT TO DECRYPT AND WHY:
TRANSPORT PROTOCOLS, APPLICATION PROTOCOLS, AND AUTHENTICATION PROTOCOLS

The discussion of whether and how to decrypt data for analysis is complex, in part because there is such a wide range of encryption systems and protocols, and such a wide range of different types of data being encrypted. This section will cover encryption schemes and decryption implications for industry protocols such as SSL/TLS, as well as Microsoft protocols—some used for authentication and some for application traffic.

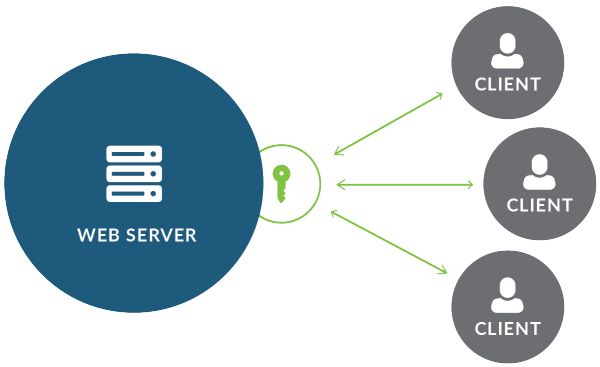
Industry Protocols (SSL/TLS)

Long Term Private Keys vs. Ephemeral Session Keys

In March of 2018, the IETF ratified TLS 1.3 as the new standard encryption protocol for network communications. The most impactful aspect of this update is the requirement of perfect forward secrecy (PFS). Previous versions of TLS allowed the use of the now-deprecated RSA ciphers for key exchange, and allowed servers and clients to use long-term private keys from which individual session keys could be derived. This meant that if the private key for a server or client was compromised at any point, all of that device's communications over the period of time the key was in use would be vulnerable to malicious actors.

RSA Key Exchange

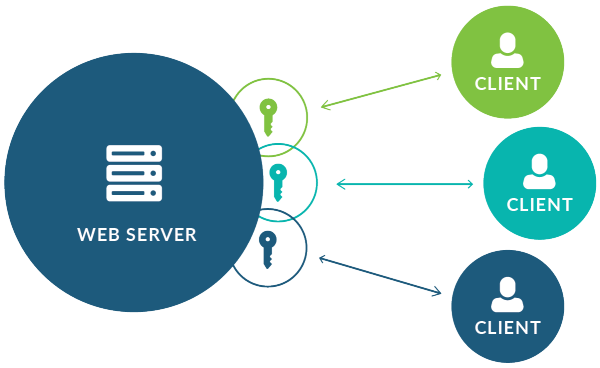
Long-Term Private Key



RSA key exchange, now deprecated, used a long term private key that meant if the key was compromised, many communications over a long period of time could be decrypted.

Perfect Forward Secrecy

Unique Ephemeral Key per Session



With PFS, standard in TLS 1.3, every session is encrypted with a new ephemeral session secret, so that any compromised key can only decrypt a single session.

This fundamental vulnerability, along with a variety of other serious vulnerabilities, led to the deprecation of TLS 1.0 and 1.1 by the IETF in March of 2021. The new standards, TLS 1.2 and 1.3, utilize Perfect Forward Secrecy (using Elliptic Curve Diffie-Hellman Encryption), which creates an ephemeral session key - or "secret" - each conversation. The ephemeral secret is only used for that conversation, and cannot be derived from the private key of either the server or the client. Even if an attacker compromised a session secret, it would only decrypt that session. Other sessions with the same server would still be secure. For hackers trying to steal large databases of intellectual property or millions of credit card numbers, this presents a significant challenge - which is the goal of cybersecurity.

Unfortunately, the same challenge is presented to SecOps teams who need visibility into their traffic in order to detect and investigate threats. This challenge is not limited only to TLS 1.3. Any environment with Perfect Forward Secrecy (PFS) enabled, regardless of TLS version, will potentially experience this loss of visibility.

Authentication protocols are popular attack vectors as they can help attackers gain access and elevate privileges.

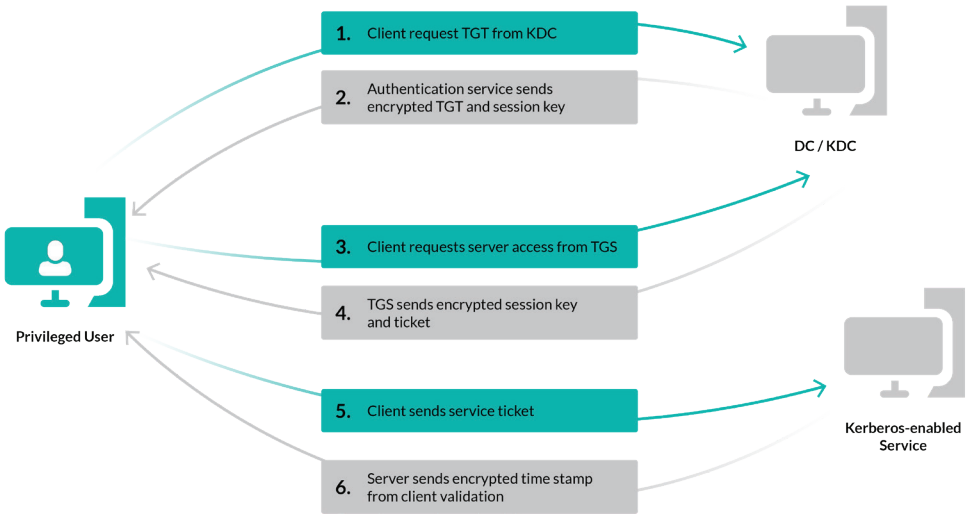
Reveal(x) helps address the visibility issue by focusing on decrypting traffic to and from an organization's public-facing services, such as email, web, and DNS servers. This approach restores visibility into critical business services allowing for the early detection of exploitation attempts (such as those associated with the ProxyShell and ProxyLogon vulnerabilities disclosed in 2021).

Microsoft Authentication Protocols

Authentication protocols are designed, as the name implies, to facilitate authentication of systems on a network. These protocols are popular attack vectors as they can help attackers gain access and elevate privileges. In the Microsoft world the most common authentication protocols are NTLM and Kerberos.

Kerberos

Kerberos is one of the oldest authentication protocols in existence. It has been heavily used by Microsoft for Authentication purposes for decades. Developed at MIT in the 1980's, it became an IETF Standard in 1993. Kerberos, so named as a reference to the three-headed dog from ancient Greek mythology, uses a three-way authentication mechanism that inserts a trusted third party called the Key Distribution Center (KDC) into the authentication process. This KDC breaks down into two logical services called the Authentication Server (AS) and the Ticket Granting Server (TGS). The fundamental concept is to eliminate the need to send passwords over the network; instead, a hash of the user's password is sent and checked on both sides of the connection.



Over the years Kerberos has undergone several revisions, with the most current being Version 5 (Release 1.18.3), released in November of 2020. Early versions of Kerberos used the now-defunct 3DES encryption algorithm, which was later replaced by the Advanced Encryption Standard (AES), as discussed in RFC3962. In 2008, shortly after the public release of TLS 1.2, the IETF began exploring the idea of using TLS 1.2 to enhance Kerberos security, and though they do not come enabled by default, those enhancements can be utilized if desired.

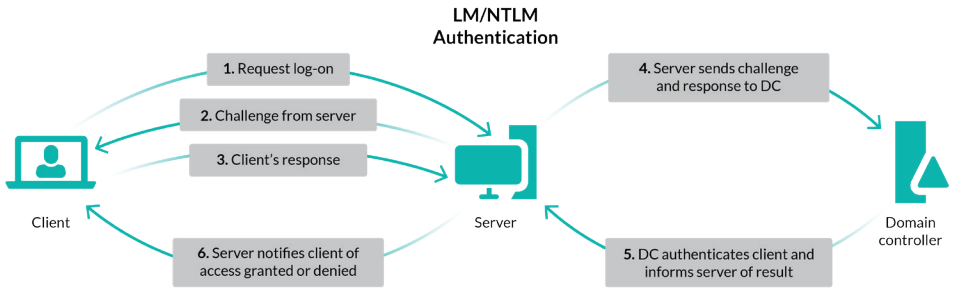
Attackers have continued to target Kerberos directly as a vector for theft or forgery of authentication material. Tactics such as Kerberos Golden Ticket Attacks, Silver Ticket Attacks, and Pass-the-Ticket have been used as privilege escalation mechanisms by savvy attackers. For SecOps teams looking to secure vital infrastructure, the decryption and parsing of Kerberos traffic offers a new level of visibility and security monitoring.

NTLM

Introduced in 1993, NTLM is an upgraded version of its predecessor, LAN Manager (or LM). First released with Windows NT 3.1, NTLM introduced the concept of a domain controller, which kept the password hashes for all users in a domain. NTLM has several advantages over its predecessor, including encrypted storage of user passwords, transmitting only a portion of a user's password, and the ability to create user security tokens that enable both authorization and authentication.

NTLM is particularly vulnerable, as it was not designed for and does not support modern cryptographic methods.

Several flaws in NTLMv1 resulted in Microsoft releasing NTLMv2 with the release of Windows NT 4.0 SP4 in 1996.



Unfortunately, NTLM is particularly vulnerable, as it was not designed for and does not support modern cryptographic methods such as AES or SHA-256. This leaves NTLM vulnerable to brute force attacks, relay attacks such as PetitPotam, and the well known pass-the-hash attack. Attacks targeting NTLM are a common tactic with many canned attacks provided by free penetration testing tools such as Kali Linux. Decryption of NTLM allows ExtraHop to monitor all NTLM traffic and identify anomalous behaviors and both known and unknown attacks against NTLM, ensuring high quality early detection of malicious behaviors.

Microsoft Application Protocols

Application protocols define how application processes (client and servers) communicate with each other over the network. These protocols are popular attack vectors as they are often ignored by security tools and many of these protocols have optional encryption.

LDAP

LDAP (Lightweight Directory Access Protocol) is an open-source, cross-platform protocol that provides the communication language that applications use to communicate with other directory services servers. It is heavily utilized in Microsoft Active Directory as a means of allowing applications and operating systems to communicate with the domain controllers. Developed at the University of Michigan between 1993 and 1997 and adopted as a standard by the IETF in 1997 LDAP has undergone three major revisions during its development, with LDAPv3 being the most current. In March 2003, the IETF released RFC 3494, which deprecated LDAPv2.

MS-RPC

MS-RPC (Microsoft Remote Procedure Call) was used in the July 1993 release of Windows NT to facilitate the client/server model. Also known as a function call or a subroutine call, MS-RPC is a protocol that uses the client-server model in order to allow one program to request service from a program on another computer without having to understand the details of that computer's network. MSRPC was originally derived from open source software DCE/RPC, but has been further developed and copyrighted by Microsoft.

SMBv3

SMBv3 (Server Message Block version 3) is a communication protocol used for sharing access to files, printers, and serial ports between nodes on a network. The SMB protocol was designed at IBM in early 1983 as a network file system protocol. Microsoft, after considerable modification, merged SMB with the LAN Manager product around 1990. SMBv1 was deprecated in 2013 with the release of Windows Server 2012 R2. SMBv2 was released with Windows Vista in 2006, and SMBv3 was introduced with Windows 8 in 2012. This version introduced significant improvements that added both functionality and better performance. Finally, SMBv3.1.1 was introduced with Windows 10. This current version added, among other things, AES-128 GCM and AES-128 CCM encryption.

Decrypting Microsoft Application Protocols

As the threat landscape shifts towards the increased use of living-off-the-land techniques, it is common to find attackers leveraging native Microsoft Protocols, such as LDAP, MS-RPC, and SMBv3, as a means of identifying additional network systems and user accounts, as well as transferring files and executing commands on remote systems. Without visibility into these protocols, the limited logging performed by default on Microsoft Windows workstation and Server platforms leaves defenders with a difficult challenge of piecing together attacks based on very limited data.

With decryption and full protocol parsing, Reveal(x) is able to detect attacks like PrintNightmare and PetitPotam without regard for the transport protocols or whether or not encryption was utilized.

HOW TO DECRYPT TRAFFIC FOR ANALYSIS: A TALE OF TWO METHODS

There are two models for accessing and decrypting data for security analytics:

1. In-band decryption
2. Out-of-band monitoring and decryption

The in-band decryption model requires placing a device in-line on the network so that all messages passing across the network are captured, decrypted, analyzed, then re-encrypted and sent along to their final destination. Though this model is widely used, research has shown that it introduces more security risks than it solves. Because this model decrypts data in-line, it must temporarily store cleartext data, making that data a juicy target for hackers. Research also shows that as much as 60% of the re-encryption methods use a weaker ciphersuite than the original message. Additionally, this model inherently introduces network latency, and none are architected to perform well at the scale and throughput levels required by today's enterprise networks.

Research has shown that the in-band decryption model introduces more security risks than it solves.

Therefore, the out-of-band monitoring and decryption method is preferable for SecOps teams monitoring east-west traffic for hidden threats. Out-of-band solutions acquire a copy of network traffic from a network tap or port mirror. Since they're not preventing the original communications from going through, they do not introduce any network latency, nor do they need to re-encrypt the communications, which eliminates the risk of using lower-quality encryption algorithms.

HOW EXTRAHOP REVEAL(X) OUT-OF-BAND DECRYPTION WORKS

ExtraHop Reveal(x) is an out-of-band solution that conducts all decryption and analytics "on box." This means it never needs to send any cleartext data across the network, nor does it need to re-encrypt any messages.

Data Acquisition

For hardware-based out-of-band solutions, acquiring data via a network tap or port mirror is a fairly straightforward process. Reveal(x) appliances can ingest, decrypt, and analyze up to 100 Gbps of traffic in real time. In cloud environments, Reveal(x) uses either Google Traffic Mirroring or Amazon VPC Traffic Mirroring to acquire the packets.

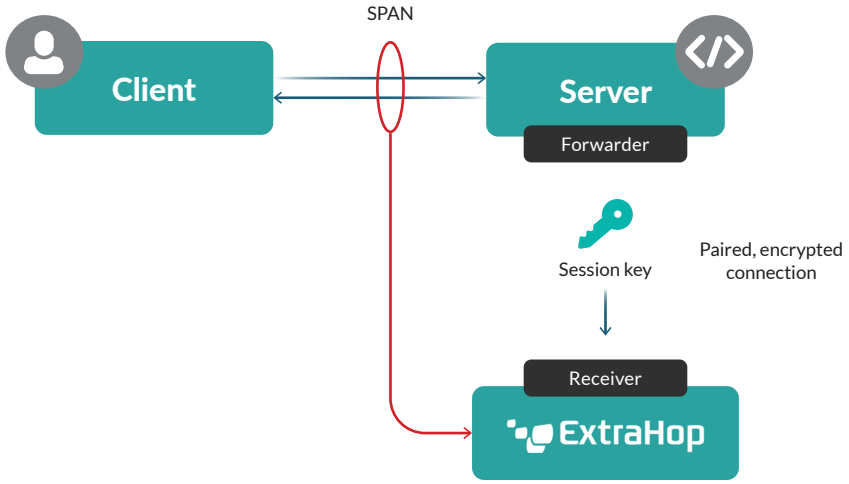
Taking Advantage of Decryption While Still Protecting Sensitive Data

Reveal(x) is designed to provide users with deep, meaningful network traffic analysis while protecting the privacy of sensitive data, personal identifiers, or data regulated by various industry standards. Reveal(x), allows customers to choose which traffic to apply decryption to

without exposing the data. The platform provides customizable controls for data access using Application Inspection Triggers and Role Based Access Controls (RBAC), so SecOps teams can get the visibility they need while staying fully compliant.

Using and Protecting Your Private Keys in TLS 1.3

Reveal(x) accesses the ephemeral session secrets for each conversation with a lightweight secret-sharing agent installed on each server. The agent securely transmits session secrets from the server across a PFS encrypted channel to the Reveal(x) appliance, where they are securely stored and only accessible to users with the highest level of administrative privilege.



An Important Note on RSA Key Exchange

It should be noted that as of TLS 1.3, RSA key exchange is deprecated. Reveal(x) still allows users to upload RSA keys, because many enterprise systems still use earlier versions of SSL/TLS. This is considered an insecure practice, and we recommend eliminating use of RSA and adopting TLS 1.3.

Accessing Critical Data with Need-To-Know Decryption

Normally, you can get all the information you need for incident investigation and response from the metrics provided by Reveal(x) without needing any person to lay eyes on unencrypted data. However, sometimes seeing the packets themselves is the only way to prove exactly what happened. Whether you're proving to a third-party vendor that their action constituted an SLA violation or providing evidence of regulatory compliance, sometimes you need access to cleartext packets.

Reveal(x) is able to provide highly granular, role-based access to the decryption keys for specific sessions. We've covered how the data and PFS session keys are acquired in earlier sections. Here's what the experience is like for individual users:
Reveal(x) users may be assigned one of three levels of access:

1. No Access
2. Access to Packets Only
3. Access to Packets and Secrets

Users with access to packets and secrets will see a new "Download Session Keys" button when looking at packets in Reveal(x). This will enable those users to download the asymmetric key to decrypt the packets transmitted between the specific clients, during the specific time window of their search. The nature of asymmetric key encryption means that the keys accessible by highly-privileged Reveal(x) users can only decrypt the exact packets the user selects. Even if the asymmetric key was compromised, it could not be used on anything beyond that narrow range of packets.

Diving Deep with WireShark

While Reveal(x) uses its decryption capabilities to provide the richest data for real-time analysis and metrics, and to provide data for machine learning behavioral detection, the product does not provide the capability, on-appliance, to manually examine individual packets that have been decrypted using PFS session keys. To decrypt and examine downloaded packets, users with the highest level of privilege need to download the session keys and the relevant PCAP files and use Wireshark to open and examine them.

IS DECRYPTION NECESSARY FOR DETECTION AND INVESTIGATION?

Many vendors of monitoring and analytics products make the claim that it is unnecessary to decrypt traffic for analysis. They believe SecOps teams can get enough information out of limited data such as NetFlow and log analytics, or by analyzing still-encrypted traffic. For the reasons listed in this brief, they are wrong. Decrypting and analyzing packets all the way down to the application transaction payload at Layer 7 frequently provides a level of definitive insight in a way that simply isn't possible with encrypted data limited to L4 flow communications. This insight is vital for SOC analysts to prioritize their actions and respond confidently to incidents before damage is done. If you want to limit the blast radius of an attack, you have to decrypt the data.

What is Encrypted Traffic Analysis and Does It Work?

This one is a little more complicated. When vendors say "encrypted traffic analysis," they often mean that they are inferring malicious behavior by looking at the sequence of packet lengths and times (SPLT) in observed transactions. For example, after an adversary compromises a machine inside the target network, they are likely to try to move laterally, exploiting vulnerabilities like PrintNightmare, compromising user accounts, and locating ways to find and access databases containing valuable data. Encrypted traffic analysis would see the related database traffic, and (possibly) infer that the cadence of the compromised machine's interactions with the database doesn't look the same as usual interactions with that database. Alternatively, encrypted traffic analysis may entirely miss attempts to exploit vulnerabilities like PrintNightmare due to the minimal amount of traffic needed to attempt the exploit. The SPLT approach lacks the requisite contextual details needed for a thorough analysis of the threat, leaving SecOps personnel spending valuable time pivoting to other information sources to confirm the detection and identify relevant remediation actions.

Encrypted traffic analysis may entirely miss attempts to exploit vulnerabilities like PrintNightmare due to the minimal amount of traffic needed to attempt the exploit.

In contrast, a product that was decrypting this traffic and inspecting the payload itself would be able to see whether the actual methods being used looked malicious. For example, seeing a series of SELECT* methods followed by a DROP TABLE would be a much clearer signal of malicious activity than a change in timing or volume of transactions. Decrypting traffic for analysis is often the only way to confidently differentiate legitimate use of a protocol from malicious tunneling by an attacker who is living off the land.

What about TLS Fingerprinting? Don't JA3 Signatures Work?

TLS fingerprinting and JA3 signature-based analytics can provide visibility - albeit limited - into encrypted traffic where decryption cannot be deployed. JA3 signatures are able to discover when new applications show up on your network, and also when a novel application starts communicating with a new endpoint. The combination of JA3 and JA3S is particularly good for detecting stealthy command & control (CnC) traffic of known malware variants. However, it lacks the ability to flag unknown signatures, which is a necessary function when spotting new malware and zero-day attacks. As such, this approach of analyzing encrypted traffic can provide a valuable puzzle piece, but not a complete picture. Reveal(x) supports JA3 and JA3S fingerprints for all TLS traffic, and also provides real-time TLS decryption for critical assets, even when PFS is used. In other words, you would have complete end-to-end visibility, investigation, and forensics.

The benefit to the encrypted traffic analysis approach is the ability to monitor sensitive network segments that are subject to compliance requirements which disallow the use of decryption, such as PCI-DSS and HIPAA. Which is why Reveal(x) still performs analysis and can detect threats in traffic that must remain encrypted for regulatory reasons.

ExtraHop Reveal(x) is the only network traffic analytics product capable of decrypting Microsoft protocols and TLS traffic at line rate at sustained 100 Gbps of throughput to provide complete visibility, real-time detection, and guided investigations about the things that matter most to the SOC.

ExtraHop Reveal(x) 360 is the only network detection and response technology that is able to decrypt network traffic out-of-band, at enterprise scale, delivering complete visibility, threat-detection, and threat remediation that is 84% faster than other leading tools. ExtraHop Reveal(x) 360 decrypts network traffic, out-of-band, at line rate, and without impacting network performance. It is the only NDR solution that can decrypt TLS 1.3, as well as the two most widely used Microsoft authentication protocols, NTLM and Kerberos. To ensure complete visibility Reveal(x) 360 is also designed to decrypt the most commonly abused Microsoft application protocols such as MS-RPC, LDAP, and SMBV3.

ADDITIONAL RESOURCES

Learn More About Why SecOps Needs Decryption to Succeed

Blog Series:

Unpacking The Looming Challenge of Encryption for SecOps, Parts 1 & 2

Blog Post:

What is Perfect Forward Secrecy?

Video:

How Does ExtraHop Perfect Forward Secrecy Decryption Work?

Already A Customer and Want To Get Started?

Here are some handy links to ExtraHop documentation about how to get started with decryption in ExtraHop Reveal(x) Network Traffic Analytics:

Admin UI Guide to SSL Decryption

Perfect Forward Secrecy Installation

Installing PFS Forwarder on F5

ABOUT EXTRAHOP NETWORKS

ExtraHop is on a mission to stop advanced threats with security that can't be undermined, outsmarted, or compromised. Our dynamic cyber defense platform, Reveal(x) 360, uses cloud scale AI to help enterprises detect and respond to advanced threats—before they can compromise your business. With complete visibility from ExtraHop, enterprises can detect intrusions, hunt threats, and investigate incidents with confidence. When you don't have to choose between protecting your business and moving it forward, that's security, uncompromised.

ExtraHop

info@extrahop.com

www.extrahop.com