Detect. Decrypt. Deter.

MIRA WHITEPAPER

Without Visibility, There is No Security

These days all network traffic is encrypted making the job of security tools significantly more difficult as threats are hidden inside the encrypted traffic flow.

This paper looks at the reasons why visibility into encrypted traffic may be required by an enterprise and then considers the technical mechanisms available to provide such visibility to security tools as well as network performance management tools. The benefits and drawbacks of different approaches are discussed with examples of what an enterprise deployment may look like.

Document version 1.0
January 2021

MiraSecurity.com

mira SECURITY

info@mirasecurity.com

Table of contents

1 Introduction

These days all network traffic is encrypted making the job of security tools significantly more difficult as threats are hidden inside the encrypted traffic flow. While encryption offers enhanced security and privacy to the end user it raises serious issues for enterprise security teams tasked with protecting the organization and ensuring that relevant legal and regulatory requirements are met. Getting the correct balance between security and privacy in the Enterprise is a challenging task that in most cases requires visibility into at least some of the encrypted traffic.

In dealing with the risks that are created by encrypted traffic, enterprises also have to cope with other developments in how the enterprise network is deployed and managed. The move to higher bandwidth connectivity within enterprise data centers and growth in the use of virtual overlays such as VLANs and tunnels such as VXLAN or GRE all have an impact. Migration of enterprise network elements to private or public cloud environments significantly affects enterprise security architectures.

This paper looks at the reasons why visibility into encrypted traffic may be required by an enterprise and then considers the technical mechanisms available to provide such visibility to security tools as well as network performance management tools. The benefits and drawbacks of different approaches are discussed with examples of what an enterprise deployment may look like.

Before looking at how encrypted traffic, specifically TLS traffic, can be decrypted to provide visibility for network security tools it is important to make clear that TLS is a very secure protocol that has evolved over time to provide enhanced security and privacy for end to end communications. The protocol prevents any third party from being able to intercept and decrypt the data enabling secure e-commerce and private web transactions. What this paper looks at is how TLS traffic can be made visible to network security tools when the enterprise that controls the security tools also has control over one of the endpoints of the TLS session - either the client or the server. None of the mechanisms discussed in this paper can be used by a third party that has no control over the endpoints of the TLS session.

Strong enterprise security relies on both endpoint security and network security. Endpoint security software based on the client or server has access to the decrypted contents of encrypted network flows so it can detect threats that were encrypted. However, relying on endpoint security alone creates risk as any compromise of the endpoint security or endpoint systems without the software installed will go undetected. Network security provides additional security that will identify threats to/from systems with compromised or non existent endpoint security. Providing visibility to network security tools is an important element of a comprehensive enterprise security strategy. Enterprises also need to address new encrypted traffic types such as QUIC which create the same issues as TLS does. While this paper focuses on TLS traffic we will touch on how QUIC traffic can be handled today and in the future.

2 Security implications of encrypted traffic

Encrypted traffic flows can be categorized into two groups:

- Traffic that originates from a client system that is not under the control of the enterprise and which terminates on a server under the control of the enterprise. For example traffic from a bank customer to the bank's Internet banking service. We will term this "enterprise server" traffic (ES).
- Traffic that originates from a client system that is under the control of the enterprise that terminates on a server that is under the control of a third party. For example an enterprise employee accessing an Internet service such as personal webmail or cloud storage. We will term this "enterprise client" traffic (EC).

2.1 Enterprise destination (ES) traffic threats

In the case of ES traffic the security risks that may be hidden inside the encrypted flow are primarily attacks against the enterprise services and the infrastructure providing it. As the encrypted flow is destined for a server in the enterprise, where it will be decrypted, there are no real privacy issues involved in this case as the enterprise has the right to see the unencrypted traffic from the customer. Typical network security tools that may be used to detect and prevent threats in this traffic are IPS/IDS, Malware detection, packet capture and forensics as well as non security focused tools related to traffic analysis and network engineering use cases.

2.2 Enterprise origin (EC) traffic threats

EC traffic can enable security risks related to exfiltration of sensitive enterprise data as well as attacks that seek to compromise internal client systems as a bridgehead to deeper penetration of the network. EC traffic also raises privacy issues when employees access personal services such as banking or healthcare sites while at work. Typical network security tools that may be used to detect and prevent threats in EC traffic include IPS, Malware detection, DLP , Web filtering, packet capture and analytics.

3 Enabling visibility into encrypted traffic

In order to understand the different mechanisms used to gain visibility into TLS traffic some basic elements of the TLS protocol need to be understood.

There are a number of different versions of the TLS protocol which developed from the early SSL protocol work. At this point in time all SSL protocol versions and TLS 1.0 and 1.1 are either formally deprecated or no longer supported by current web browsers or Internet services. While the vast majority of TLS traffic will be either TLS 1.2 or 1.3 there will still be some traffic that for

various reasons is still using earlier versions of TLS or SSL so providing visibility into these versions is still needed.

All versions of TLS prior to TLS 1.3 operated in similar fashion. TLS 1.3 involved major changes in how the TLS protocol works and this means that providing visibility into TLS 1.3 traffic requires different mechanisms to those used for earlier versions.

3.1 TLS details that impact visibility solutions

A TLS session involve the following steps:

- Establishment of a TCP connection between a client and a server.
- A TLS handshake phase during which the client and server negotiate the parameters the TLS session will use and agree on the cryptographic keys that will be used during the session.
- A data transfer phase during which end user data is exchanged between client and server in encrypted form.
- Termination of the TLS session and the underlying TCP session.

All TLS 1.3 sessions and any TLS 1.2 sessions using best practice will use Ephemeral Diffie Helman (DHE) key exchange during the handshake in order to provide "perfect forward secrecy" (PFS). PFS provides protection against replay attacks in the event that the server's private key is compromised. Without PFS the encrypted TLS session can be recorded and then replayed to a system using the private key from the server and it can be decrypted. With PFS even with the server's private key it is impossible to decrypt the recorded session.

In order to decrypt the flow the decryption device needs to have the session key being used for this flow. The session key is known to the endpoints as they negotiate it during the TLS handshake but is not exposed on the wire so it cannot be obtained by simply sniffing the packets. Any device that is going to decrypt a flow to feed security tools needs the session key and either has to obtain this from one of the end points or become a controlled "man in the middle" (MITM). As a MITM the device can participate in the TLS handshake and participate in the session key negotiation with the result that it ends up knowing the session key being used by both the client and the server.

During the TLS handshake the client and server negotiate parameters that affect the security profile of the session. These include:

- Which version of TLS is to be used.
- Which cipher suite is to be used. The cipher suite determines whether PFS is being used, what key types and sizes are in use, what bulk cipher is used to encrypt user data as well as other parameters.

The server also sends a server certificate to the client which the client can use to validate that it is talking to the correct server. If the server certificate is issued by a publicly trusted CA then the client will trust it. If the server certificate is issued by an untrusted CA or is self-signed then the client software will normally generate warnings or prevent the session from proceeding.

When inline visibility is being used the visibility appliance participates in the TLS handshake with both the client and the server. This means that it can influence what parameters are negotiated for the end to end encrypted flow. Mira takes the view that a visibility appliance should, whenever possible, be "transparent" meaning that the negotiated parameters during the TLS handshake should be the same when the visibility appliance is doing inline decryption as they are when it is not present. This means that the act of providing visibility will not downgrade the security profile of the end to end session, not all visibility solutions take this approach.

3.2 Provide visibility or not?

An important part of any visibility solution is the policy control provided to the enterprise allowing them to determine which encrypted flows are made visible and which are not. Typical policy triggers include:

- Packet header details such as IP address, port number, VLAN ID etc.
- TLS Client Hello details such as SNI, supported TLS versions, supported cipher suites etc.
- TLS Server response details such as selected TLS version, server certificate details etc.
- External data such as the category the destination server is classified as belonging to.

If the triggers in a policy rule are matched then the action taken can be:

- Leave the flow alone so it remains encrypted and the network security tools will see the encrypted flow,
- Provide visibility into the flow using the appropriate decryption mechanism and send unencrypted data to the network security tools.
- Prevent the flow from happening by either dropping the packets or sending a TCP reset. This might be used to prevent deprecated versions of TLS from being used for example.

Being able to use category information in policy decisions is a significant feature that makes aligning visibility policy with corporate privacy and security policies much simpler. For example a policy could determine that any EC traffic to financial service (banking) sites or to health care sites was always left untouched in order to protect employees' privacy.

3.3 Passive (out of band) decryption

A common assumption is that visibility into encrypted traffic can be done out of band (OOB) by looking at copies of the packets from a span or mirror port. In most situations this is not possible and the visibility device will need to be inline with the original packets passing through it.

The two situations where OOB visibility is possible are:

- When the session key for a flow is obtained from one of the endpoints and made available to the visibility device. This requires that special agent software is installed on the endpoint and a secure real time key distribution mechanism exists to get the session key from the end point to the visibility device before it sees encrypted packets. While either end point could provide the session key in reality this approach is only feasible when the

server provides the session key. Obtaining the session key from the server restricts this approach to providing visibility into ES traffic. One issue with this approach is that it requires modification to the server to install software that will extract and distribute the ephemeral key. In regulated environments such modifications to secure servers may not be allowed or may require protracted certification activity. Ensuring the secure distribution of the ephemeral keys may also require certification or regulatory approval.

- When a TLS session does not use PFS then OOB inspection is possible for ES traffic as long as the visibility device has a copy of the server's private key. As noted earlier best practise is to use PFS so the amount of traffic that this technique will work for is minimal. There are some special cases where within a data center internal traffic may not use PFS in order to allow OOB decryption but in general this approach is not widely used.

3.4 Active (inline) decryption

The vast majority of visibility into encrypted TLS traffic for network security tools is achieved using inline decryption where the visibility appliance is a "bump in the wire" with the actual packets for the encrypted flow passing through the device. A security tool may be its own visibility device with the decryption being carried out within the tool or a separate visibility tool may focus on decryption with the ability to feed decrypted flows to attached security tools.

Security tools that are able to block threats or modify data in order to mitigate risk need to be inline and able to modify the actual data within the flow while it is decrypted. Such tools can only work with a decrypted flow if the visibility device is carrying out inline decryption. An inline visibility solution can feed both inline security tools and passive security tools that simply consume data.

There are a number of different mechanisms used for inline decryption depending on the type of traffic being decrypted. The two main mechanisms are:

- Known server key to decrypt ES traffic flows. In this mode the visibility device has a copy of the server private key and uses that to participate in the TLS handshake. The device knows the session keys in use for the TLS session between itself and the client and itself and the server allowing it to decrypt and re-encrypt the traffic to maintain an end to end encrypted flow.
- Certificate resign is used to decrypt EC traffic flows. In this mode the visibility device modifies the server certificate and signs it with an internal Certificate Authority (CA) before sending it on to the client. The client has to trust the internal CA being used by the visibility appliance otherwise client software such as browsers will generate warnings or even prevent the session from working.

It is worth saying that there is nothing preventing an inline visibility appliance using session keys received from one of the endpoints to decrypt the flow. Such an approach would work to decrypt EC and ES traffic. Allowing such a device to support inline tools that modify the decrypted traffic flow would be complex but possible however the issues mentioned earlier in relation to obtaining session keys from endpoints still apply.

3.5 Third party visibility (cloud security service)

Security as a Service providers offer cloud based security solutions that allow an enterprise to route all or some of their traffic through the cloud service so that network security applications that are part of the service can provide protection. Such services often include the ability to provide visibility into encrypted flows for the security applications allowing the service to protect against threats within encrypted flows.

There are issues for enterprises when using third party cloud security services to deal with threats within encrypted traffic that may prevent or restrict the extent to which these services can be used . Some example issues are:

- If a financial service provider were to use a third party cloud service to provide security against threats in ES traffic then this would require that the third party is provided with the private keys of the enterprise servers offering financial services, in many cases this would not be allowed by either regulatory or corporate policy. If sharing private keys was allowed there would still be the issue of a third party having access to the unencrypted details of a customers online banking session which may require specific customer buy in to avoid breaching privacy regulations.
- In the case of both ES and EC traffic the security profile for a user's session is determined by the destination server. When inline decryption is being done by a visibility appliance the appliance participates in the TLS handshake and determines what the security profile is for the end to end encrypted session. If the third party does not support the TLS version, cipher suite or key type/size that the security profile would normally use then the session may have a lower security profile when it is subject to inline decryption in the cloud. This issue of potential weakening of the security profile for a flow also applies in the case where the enterprise is responsible for inline decryption but in this case the enterprise will be aware there is an issue and has the option of addressing it by using a more capable inline decryption solution.
- If a third party cloud solution is being usd to provide security for enterprise EC traffic then the CA that is used to sign modified server certificates to allow inline decryption may be owned and issued by the third party cloud security provider. In this case the enterprise client machines have to be configured to trust the third party CA but the enterprise does not control the use of the CA or have the ability to revoke it if needed. If the cloud solution allows the enterprise to provide an enterprise issued intermediate CA then the enterprise has more control but may encounter issues with providing the intermediate CA and private key to a third party.

4 Deploying visibility solutions within the Enterprise

4.1 Bump in the wire

An inline visibility solution is commonly deployed as a “bump in the wire” meaning it has two interfaces connected to the network through which traffic passes. These interfaces are transparent in the sense that they do not have IP addresses associated with them so deployment does not require any changes to the existing network addressing or IP routing. Depending on the type of security appliances attached to the visibility device there will be at least one or two additional interfaces. A single interface is sufficient to feed decrypted data to a passive security tool, though more may be used to either feed multiple tools or to load balance traffic to a single tool. An active security device, such as an IPS, needs to be inline with the end to end traffic flow so the visibility appliance will need two interfaces to connect to it.

It is quite common for a visibility appliance to need to feed both an active security tool and one or more passive tools so additional interfaces may be needed to support this. If a large number of interfaces are required to feed multiple passive security tools then it may be more efficient to use a single interface on the visibility device and to replicate and distribute the traffic using a packet broker.

4.2 Bump in the tunnel

Many enterprise data centers have collapsed their networks down to a smaller number of higher capacity links, for example moving from 10Gbps links to 25Gbps or 40Gbps links. As part of moving to higher capacity links it is common to create virtual overlay networks using VLANs or tunnels such as GRE or VXLAN in order to simplify management and configuration of the network. Deploying a simple bump in the wire visibility appliance in a network work with a virtual overlay can be problematic as the device may not be able to handle VLANs or tunnels transparently forcing the overlay to be terminated and then re-created either side of the visibility appliance.

A visibility appliance that supports “bump in the tunnel” operation understands VLANs and tunnel protocols and can detect encrypted flows within the tunnels and decrypt them if required. As the tunnels are not terminated by the visibility device the tunnel support is transparent in the same way that bump in the wire support is transparent. A bump in the tunnel appliance can be deployed without requiring any changes to the existing overlay network.

4.3 Private cloud

Enterprises that have moved from physical to virtual data centers hosted in private cloud environments need to replicate the security infrastructure that was present in the physical data center. This creates a requirement for virtual visibility appliances that can exist in a private cloud

and feed decrypted flows to virtual network security tools that are also resident in the private cloud.

As long as a virtual visibility appliance can operate in " bump in the wire" and "bump in the tunnel" modes it can be used in the private cloud environment. As virtual overlay networks using tunnels are common in private cloud environments this mode of operation is more important here than in a physical data center deployment.

A virtual visibility appliance that supports bump in the wire/tunnel can also be used in the physical data center, perhaps as an initial deployment that is then transition ready for when the data center migrates to a private cloud.

4.4 Public cloud

Providing visibility into encrypted traffic for enterprise security tools running in public cloud environments is more complex than for private cloud for a number of reasons. In addition actions need to be put in place to cope with the fact that this is a public environment not a private one so extra security is required.

The first difference with public cloud environments is that a transparent layer 2 bump in the wire or bump in the tunnel is not normally supported. This means that a visibility appliance will require layer 3 addressing for all it's virtual interfaces. Tunnel support that can terminate and originate tunnels rather than just be transparent to them is also likely to be required.

While a visibility appliance will not typically collect or store the decrypted traffic the attached security tools may do this and this may be seen as a security risk by an enterprise. Mitigating this risk requires security features within the security tools to protect the data they collected rather than any action by the visibility appliance.

Public cloud environments do have features that are useful for visibility appliance deployments:

- Most provide some form of virtual tap capability which can be used to send decrypted traffic from the visibility appliance to passive security tools. One issue to consider here is that the decrypted traffic is passing over the public cloud between the visibility appliance and the network security tool which is a potential risk. This can be mitigated by using an encrypted tunnel between the visibility device and the security tool(s).
- Virtual HSM services are provided by some public cloud providers and can be used to securely store enterprise certificates and keys that need to be used by a visibility appliance. Some customers may be prevented from storing server certificates and keys and enterprise intermediate CA certificates in the cloud, particularly in areas like finance and healthcare. If a customer is required to keep certificates and keys on premise in an HSM then the visibility appliance in the public cloud will require a secure way to access and use these.

5 Summary

Network security tools are an important part of an enterprise security strategy. Encrypted traffic can prevent these tools from functioning as threats they could detect/block are hidden inside the encrypted stream. Providing visibility to multiple network security tools from a visibility appliance enables these tools to function effectively as they will receive unencrypted data. Providing visibility to the security tools requires that the enterprise has either control of the client system or the server system involved in the TLS session.

A safe and secure visibility system will provide transparency in a number of contexts:

- Deployment transparency. No need to reconfigure network addressing, routing, overlay configuration etc. when deploying an inline visibility solution
- TLS transparency. The TLS version, cipher suite, certificate type etc. should not be different when a flow is being made visible to when it is not. The security profile of the end to end session should be the same with or without the visibility appliance being present.

While this paper has focused on TLS encrypted traffic there are new protocols such as QUIC which offer alternative ways to provide end to end encrypted connections. Providing visibility into QUIC requires very different mechanisms to those used for TLS and while these are being developed they are not available today. Most enterprises will configure their firewalls to prevent outbound UDP traffic with a destination port of 443 which will prevent QUIC from being used and cause client software (browsers) to fall back to using TLS over TCP.

MiraSecurity.com

info@mirasecurity.com

Mira Security
3159 Unionville Road, Suite 100
Cranberry Township, PA 16066
Phone: +1 (412) 533-7830

Email: info@mirasecurity.com
mirasecurity.com

MIRA-ETO-WP-1-1/21