INTERNET-DRAFT
Intended Status: Informational
Expires: September 6, 2018

S. Fenter
Enterprise Data Center Operators

March 5, 2018

Why Enterprises Need Out-of-Band TLS Decryption
draft-fenter-tls-decryption-00

Abstract

Some enterprises are heavily TLS encrypted within their own enterprise network boundaries. Many of these enterprises are also utilizing out-of-band TLS decryption in order to inspect their own traffic for purposes of troubleshooting, network security monitoring, and for other kinds of monitoring. These monitoring functions are mission critical, and cannot just be done without when TLS 1.3 (draft-ietf-tls-tls13-26) is released or when the RSA key exchange is someday deprecated from TLS 1.2 (RFC5246). This draft will outline the use cases for out-of-band TLS decryption, as well as alternative suggestions for monitoring and troubleshooting and the limitations of those alternatives.

Table of Contents

1. Introduction

Most enterprise networks originally transmitted packet data in the clear inside their internal networks. Many still do today. When certain enterprises started TLS encrypting their internal networks to protect against insider threat and/or for regulatory compliance reasons, they always had the option of using RSA key exchanges and using static RSA private keys for a small, privileged group to decrypt and inspect their traffic out-of-band. Out-of-band decryption provides ubiquitous packet payload visibility inside the enterprise that cannot be replaced by inline/MITM decryption solutions. Today there are enterprises with extensive packet broker networks who are doing out-of-band TLS decryption to feed network sniffers, intrusion detection devices, fraud detection, malware detection, application performance monitoring tools, customer experience monitoring tools, and other solutions.

The capability to do out-of-band decryption has been available for twenty years, and for the first time in history it will be gone with the move to TLS1.3 [TLS13]. A large body of tools has grown up over the last twenty years that is dependent on out-of-band decryption. These tools are performing mission critical functions for

enterprises, and the loss of out-of-band decryption will create major operational problems for TLS encrypted enterprises if TLS 1.3 is implemented as-is inside the enterprise. Ubiquitous packet capture and decryption are required for enterprise troubleshooting, and without this capability there will be high severity outages that cannot be solved in an acceptable time frame. The outcome will be the same as extended Denial of Service attacks on enterprises worldwide. Without an out-of-band decryption solution, enterprises are left with the unattractive option of inline/MITM decryption at the data center edge and running traffic with legacy protocols or in the clear throughout the data center if they need packet payload visibility. This opens certain enterprises up to significant regulatory and insider threat problems. There are reasons why other forms of troubleshooting and monitoring do not functionally replace the visibility lost from losing out-of-band TLS decryption. These alternative suggestions are discussed in the sections below.

TLS 1.2 [RFC5246] is not a long term option for enterprises. The RSA key exchange is gradually being removed by vendors as a TLS 1.2 option. For example, mobile devices have been seen to send TLS 1.2 Client Hello's with no RSA key exchange options. There is also the risk that new vulnerabilities and weaknesses will be discovered with TLS 1.2 and/or RSA that will accelerate its removal by other vendors.

2. Out-of-Band Decryption Use Cases for Diagnostics and Troubleshooting

2.1 Application Performance Monitoring

Network-based Application Performance Monitoring requires TLS decryption in order to track application response time by user and by URL, which is the information that the application owners and the lines of business need. The end user IP address is obscured by TLS termination and by source address translation, both on the Internet and within the data center. The identity of the user can only be determined by looking at the payload of the TLS packet. URL identification allows the application support team to do granular, code level troubleshooting and performance monitoring at multiple tiers of an application.

2.2 Network Diagnostics and Troubleshooting

2.2.1 Network Packet Analysis

The key to effective network packet troubleshooting is the ability to follow a transaction through multiple tiers of an application in order to isolate the fault domain. The extensive use of both encryption and source address translation in some enterprises has made it difficult, or even impossible to follow a transaction without the ability to decrypt TLS and examine the payload of the packet. When the payload is available, the packet analyst can find unique identifiers (userids, session ids, etc.) as well as the URL the end user is accessing in order to identify the correct TCP conversation.

The packet payload allows analysts to trace the slow or failing transaction above and below firewalls, above and below load balancers, above and below switches, etc., in order to isolate the fault domain. This kind of analysis is not possible from the endpoint alone. There are firewalls and load balancers that do not terminate TLS, and there can be a significant number of infrastructure devices in between the TLS endpoints, any one of which can be the cause of a problem. This above/below analysis across all intermediate infrastructure devices often provides the only insights into where the root problem is introduced.

It is noteworthy that adding more inline/MITM decryption solutions to a multi-tiered application environment would increase the need for above/below analysis to rule in or out the inline decryption solution itself as the cause of a problem. The increased complexity of the environment would lead to more failures and longer problem resolution times.

Packet payload visibility also allows an analyst to match up a

network packet trace with log entries in an application log that give an indication of the problem that occurred. This kind of application level packet tracing could be between bare metal servers, or between two VM's on the same hypervisor. Inline/MITM TLS decryption solutions do not scale for this kind of troubleshooting. As an example, because a problem could hit anywhere, an inline/MITM solution would be needed between every possible pair of VM's in a virtual environment, which could scale into the thousands. This

large number of inline/MITM solutions would also create its own problems due to the complexity added to the environment.

2.2.2 Packet Analysis with Source Address Translation

Content Delivery Networks on the Internet have multiple TLS termination points, and the user's source (client) IP is lost. When tracing a data center's Internet connection, it is not possible to even find the correct TCP conversation for an end user that is having a problem without TLS decryption. The packets must be decrypted and the appropriate HTTP header fields examined in order to find the end user IP address.

Within the data center, the source IP for inbound TLS can again be changed multiple times. If a load balancer does not terminate TLS it will NAT the source IP so that return packets will find their way back to the load balancer, and to the correct load balancer in a pair. Alternatively, the load balancer may terminate TLS, and start an independent TLS session to the next layer below. Again, the source IP becomes the load balancer's IP address, and return packets will find their way back to the correct load balancer.

Reverse proxies, web servers, app servers, and middleware servers can all terminate TLS and start independent TLS calls to lower layers, each time altering the source (client side) IP of packets, and calling a completely different URL. User sessions are also often sprayed randomly by load balancers to all these devices, and the network troubleshooter is left with no option except to trace all packets at a particular layer, decrypt them all, and look at the payload to find a user session. Servers and infrastructure devices typically don't have the horsepower to trace and decrypt all packets like this, but an out-of-band packet broker/sniffer infrastructure is designed to handle this load, and also provides a centralized location for managing and securing capture files.

2.2.3 TCP Pipelining - Session Multiplexing

When TCP Pipelining/Session Multiplexing is used, multiple end user sessions share the same TCP connection. For the network troubleshooter, even if he/she could find the correct encrypted TCP

connection for an end user, there is no way to tell which packet

belongs to which end user without decryption.

2.2.4 TLS Encrypted MQ to the mainframe

MQ requests to the mainframe are farmed out to multiple processing nodes, and the MQ response comes back asynchronously from any one of those processing nodes. There is no way to find the response to a particular request without looking at identifiers in the payload of the packet. This requires TLS decryption. MQ requests can also pass through many infrastructure devices (i.e. load balancers, firewalls, etc.) before reaching the mainframe. Inline/MITM decryption solutions are not scalable for this environment, because visibility could be needed anywhere along the path.

2.2.5 Application Layer Data

A decrypted TLS packet contains a wealth of critical troubleshooting information for HTTP (e.g. HTTP requests and return codes) as well as for a number of other protocols. Without this level of information, network troubleshooters are blind, unless the problem is some kind of a basic network problem. Even in the case of network problems, the application level detail is sometimes critical for isolating problems, for example, in the case of an intermittent network slowdown or failure. When looking through millions of packets, transactional/application level detail can help the analyst zero in on the correct location in the trace where a network problem is occurring.

It is not enough, though, to look only at the HTTP headers. Applications have been known, for example, to return an HTTP 200 OK, yet contain an error message in the payload of the HTTP response. This can only be seen in the decrypted application layer payload of the packet.

Applications also use XML or JSON structures in the payload of the packet to store interesting information like user ids and session IDs. Oftentimes this is the level of information that the application support teams possess ("I sent out a request with this session ID and didn't get a response."). The application team doesn't, however, know where their request went wrong among the many layers of infrastructure, network connections, etc., that their request passes through. The network packet troubleshooters are able to follow the transaction through the many layers of infrastructure if they are able to access the packet payload and find a matching identifier.

2.2.6 Customer Experience Monitoring

Enterprises involved in online commerce have a business need to monitor customer behavior on their web sites. This monitoring requires TLS decryption of the full packet payload, as any location on the web page can be of interest to the user and to those monitoring user behavior.

2.2.7 VoIP Analysis

When attempting to monitor and/or troubleshoot user experience within voice and video communications, the ability to understand the signaling (session setup and teardown) is absolutely critical. Session Initiation Protocol (SIP) is among the most commonly used control protocols in the VoIP environment, and increasingly it is being encrypted with TLS.

SIP request and response codes, when visible, make it possible for even a novice user to understand the basics of what is being requested and what type of response is being provided (Request: INVITE, Response: 200 OK). The detail available in SIP messages enable a level of analysis difficult to mirror using any other method. The SIP/RTP stream must be decrypted in order to allow analysis of these setup messages. In fact, it is not possible to even find the UDP connections to analyze without decryption of the SIP header, because the IP/port pair is found in an SDP of the SIP signaling packets.

Phone endpoints are typically not designed for detailed troubleshooting. Many handsets do not have the ability to output SIP signaling information. Endpoints are also not completely trustworthy in a troubleshooting scenario, and network analysis is needed to verify what is happening on the wire. VoIP calls can also be affected by network conditions. Tracing may be done in different locations to identify the effect the network is having on the VOIP call. An inline/MITM solution doesn't scale for this use case.

Session Border Controllers can trace SIP signaling, but tracing is often too resource intensive to run on these devices, as they are not designed to handle the extra load. This means that VOIP analysts need out of band packet capture and decryption solutions which are designed for this purpose.

Call quality on the audio RTP stream can be monitored with network based tools, if TLS on the audio stream can be decrypted. This gives the VoIP analyst a view of the problem that they can't get from the

endpoints. The audio stream is peer-to-peer communication necessitating many visibility points. Again, an inline/MITM

decryption strategy doesn't scale.

2.3 Out-of-Band Decryption Use Cases for Network Security Monitoring

2.3.1 Layer 7 DDoS Attacks

Layer 7 DDoS attacks can involve multiple IPs and source-ports generating traffic very similar to that of genuine users. The only way to identify layer 7 attack traffic is via inspecting fields in the packet payload, which are invisible until the packet is decrypted.

Internet based DDoS protection services are not perfect. If they are tuned too tightly they block some legitimate production traffic. If they are tuned too loosely, some attack traffic gets through. In reality, during a DDoS attack some attack traffic usually gets through, and enterprises have to be armed to protect themselves. One of the tools they need is the ability to decrypt Internet TLS traffic so they can block layer 7 DDoS attacks.

This decryption could be done by an inline/MITM solution, although there is a possibility that an inline decryption solution could be overwhelmed by a volumetric DDoS attack or by an attack targeting session state, becoming a point of failure. An out-of-band or transparent TLS decryption solution does not carry this risk of being overwhelmed and blocking all legitimate traffic.

2.3.2 Fraud Monitoring

Fraud monitoring is the monitoring and detection of suspicious activities within, through, or perpetrated against a company. It must be reported to regulatory agencies as required by applicable laws and regulations. Examples of fraud are unauthorized account access and identify theft. Fraud monitoring is a mission critical function for financial institutions, and there are network-based tools performing this function with decrypted TLS packets. If fraud monitoring is down, then it is a severity one problem for critical applications.

Fraud monitoring looks at network packets in many locations. An

inline/MITM solution in this environment does not scale.

One of the major fraud monitoring applications consists of an array of servers, including a database, all talking to each other via TLS. Application errors for this fraud monitoring app need to be analyzed just like any other application, including network packet analysis, and TLS decryption is needed in order to match up log errors with network packets on the wire.

2.3.3 Intrusion Detection System

IDS inspection looks for known and custom malware signatures, potential attack patterns, and known observables associated to Indicators of Compromise in the payload of TLS packets when decryption is available. IDS inspection for inbound and/or internal TLS sometimes depends on out-of-band TLS decryption, and its effectiveness is severely impacted if decryption is not available.

IDS inspection is often a regulatory requirement, for example, for cardholder data environments, of which an enterprise may have many. An inline/MITM TLS decryption solution has scalability problems in this kind of environment.

2.3.4 Threat Detection and Incident Response

IDS Alerts - Threat Detection teams receive IDS alerts and will analyze decrypted network packet traces in order to verify if the alert was valid or was a false alarm.

SQL Injection Attacks - This particular alert also needs manual analysis of packet traces in order to identify if the attack was successful, and if so, what data was returned.

Endpoint Monitoring Alerts - These alerts often need to be verified with decrypted packet traces, including identification of the source of the attack on the endpoint. Endpoints are less trustworthy than network monitoring tools, and network monitoring is also needed as a backstop for any failures of monitoring on the endpoint.

Manual Hunting - Not all attacks are caught by automated monitoring. Threat Detection teams will do manual hunting for known

vulnerabilities with decrypted packet traces.

Ubiquitous packet payload visibility can be provided by out-of-band decryption for inbound or internal TLS sessions. Traffic sources for malware can be anywhere within the enterprise or external to the enterprise. An inline/MITM decryption solution doesn't scale.

2.3.5 Regulatory Examples

2.3.5.1 PCI (Payment Card Industry)

2.3.5.1.1 PCI and TLS Encryption

The PCI Security Standards Council strongly recommends segmenting the cardholder data environment in order to protect the cardholder systems as well as to limit the scope of PCI assessment (PCI

Information Supplement: Guidance for PCI DSS Scoping and Network Segmentation, p. 6). PCI has a concept of "connected to" systems, meaning that any system communicating with a system in the cardholder data environment is drawn into PCI requirements. As a practical reality, large enterprises could not get a PCI assessment completed without segmenting the cardholder data environment with tools like firewalling and encryption of data in transit. This creates the need for TLS decryption in order for those enterprises to do troubleshooting, network security monitoring, and other packet-based analysis in which the clear-text payload must be available.

2.3.5.1.2 Intrusion Detection

The PCI DSS (Data Security Standard) requires IDS/IPS inspection at the perimeter of the cardholder data environment, as well as at critical points in the cardholder data environment (PCI DSS section 11.4). If an organization's monitoring and data loss prevention strategy includes payload inspection, TLS encrypted traffic in these environments must be decrypted. PCI applications have grown up over time, and there may be many cardholder data environments in a data center. Inline/MITM decryption solutions are not scalable for this environment due to cost, introduced latency, and production risk from the more complicated, inline/MITM environment.

2.3.5.1.3 TLS 1.2 and PCI

When significant vulnerabilities were found in SSL and early TLS in late 2014 (including POODLE), it took the PCI Security Standards Council less than a year to require a migration plan away from these SSL/TLS versions (PCI Information Supplement: Migrating from SSL and Early TLS). Enterprises are at risk that vulnerabilities could be found in TLS 1.2 or in the RSA key exchange, and that PCI will require upgrade to TLS 1.3. There is no guarantee that TLS 1.2 will be available many years into the future.

2.3.5.2 e-CFR (Electronic Code of Federal Regulations)

2.3.5.2.1 Insider Abuse

The United States e-CFR, Title 12, Chapter 1, Part 21 requires that national banks and federal branches and agencies of foreign banks monitor and report on insider abuse. This monitoring looks for criminal behavior like employee fraud, and is a mission critical and legally required function for financial institutions operating in the United States. If potentially illegal or fraudulent activity is detected, a Suspicious Activity Report must be filed with FinCEN (the Financial Crimes Enforcement Network of the US Department of the Treasury).

One of the tools for accomplishing this monitoring is a network based tool that uses out-of-band TLS decryption to inspect network packets and analyze user activity. The user endpoint cannot be trusted for this monitoring, as it is controlled by the user being suspected of fraudulent or illegal activity. The loss of out-of-band decryption would be crippling to this monitoring. There are many monitoring points for this tool, and inline/MITM decryption is not a scalable option. Also, these kinds of tools have no need to be inline, and forcing the solution inline adds unnecessary complexity and production risk into a mission critical environment.

2.3.5.3 Regulatory Requirements - Summary

This section is by no means an exhaustive discussion of all regulatory requirements for all verticals in all countries worldwide. However, it does illustrate the kinds of issues that can come up when a long standing "feature" is eliminated from a network security protocol.

3. Alternative Solutions Offered and Their Limitations

3.1 Inline/MITM Decryption

This is a valid sounding option but presents scalability problems for large, diversified enterprises.

TLS decryption for security monitoring is needed in more locations than just everything in and out of the Internet; for example, network security monitoring is done on Business to Business connections, the branch/MPLS head-end, the mainframe, cardholder data environments, wireless controllers, DNS servers, etc.

Network troubleshooters need to be able to take traces and decrypt them anywhere in the enterprise network. This can be hundreds or even thousands of locations, depending on the particular problem that hits, and can include the data center, branches, the virtual environment, and public or private clouds. It's not scalable to try and put an inline/MITM decryption solution between any two servers in the enterprise who may be talking to each other via TLS. This could include VM's talking to each other on the same hypervisor or containers talking to each other on the same VM.

Cost - Bypass taps and TLS decryption appliances are expensive, and the cost adds up when adequate resiliency and failover is architected. The cost for implementing an inline/MITM decryption strategy can quickly escalate into millions of dollars.

Latency - TLS decryption appliances add 1-3 ms of latency per packet

in a hardware appliance. Virtual decryption appliances may take a larger latency hit. When multiple inline decryption locations are implemented, the latency becomes prohibitive.

Production Risk - Bypass taps and TLS decryption appliances are complicated devices that can and do fail. When an inline/MITM TLS decryption solution fails, production traffic is brought down. An inline/MITM solution is far more complex than putting in passive network taps, which are very simple and almost never fail.

Out-of-band TLS decryption is a better design for much of the

enterprise. It provides for ubiquitous packet payload visibility with lower cost, no latency, and almost non-existent production risk.

Obtaining packets in the virtual and cloud environments is more complex, but an out-of-band TLS decryption solution is still more scalable than inline/MITM TLS decryption everywhere in the virtual or cloud environment.

3.2 Using TCP or UDP Extensions to Supply Extra Information

This is also an argument that sounds good on the surface and looks like it helps preserve privacy. However, there are a number of reasons why this idea doesn't work in an enterprise.

There can be session identifiers in the payload of packets that are needed in order to match up packets (whose source IP may be changing) inside and outside of an infrastructure device, and also to match up application layer requests with application layer responses (the responses don't always ride on the same TCP conversation as the request). These session identifiers are unique to each application, and it is not possible to anticipate, among thousands of applications, which fields in the payload are going to be important for a particular problem. An individual enterprise can have thousands of unique applications. The next enterprise down the road can have thousands more applications of their own, all unique and different from the first enterprise.

Software bugs can leave telltale signs in the payload of a packet. These telltale signs are critical pieces of information for troubleshooting difficult problems. It is not possible to anticipate, among thousands of applications, which parts of the payload are going to be important for finding a software bug.

Indicators of Compromise can exist anywhere within the payload of a packet. It is not possible to anticipate for every new attack which part of the payload will be important for threat detection and incident analysis.

Fields that can be used to block layer 7 DDoS attacks can be anywhere within the payload of the packet. It's not possible to determine which field to block on for any particular DDoS attack until the full payload is decrypted and examined.

Customer Experience Monitoring requires full packet payload. A click anywhere on a web page can be of interest to those doing the monitoring.

3.3 Using IP and TCP Headers for Monitoring and Troubleshooting

This approach has all the same problems as those outlined in section 3.2 above.

3.4 TLS 1.2

No enterprise wants to run an older, less secure version of a protocol for the long term.

The RSA key exchange is already in the process of being deprecated from TLS 1.2 in some environments. Examples of this have already been seen in the mobile device environment.

Suggestions have been made on the TLS email list that we need to deprecate the RSA key exchange from TLS 1.2. All we need is for another major RSA vulnerability be found, and this sentiment will gain traction.

3.5 Logging

There are many enterprise outages and slowdowns where there is either no log message on the offending device, or there is a log message that indicates a problem but no clue as to the fault domain or the root cause. Infrastructure devices do not understand layer 7 and so are unable to log meaningful information about a layer 7 transaction that had a problem, even if that particular infrastructure device was the cause of the problem. In many cases, network packet analysis with TLS decryption is required in order to identify the fault domain and/or get to the root cause.

It is not possible for a code developer to anticipate every possible problem that is going to occur and put a log message in just the right place. Also, the very nature of a software bug is that the developer doesn't know it's there, so there is not going to be any log message when a bug hits.

It is not feasible to go through millions of lines of code in an enterprise environment and "improve" the logging on each device.

Between infrastructure and security devices, and application code, this would involve getting hundreds or thousands of vendors to invest and cooperate with this idea. Vendors would have no idea what to log other than what is currently being logged in order to try and catch the "next" problem.

Adding log messages after a problem hits is like playing enterprise "Whack-a-mole". The next problem to hit is invariably something completely different.

3.6 Troubleshooting at the Endpoint

The shortcomings of endpoint logging are covered in section 3.5 above.

Endpoints in a typical enterprise don't have the robustness to run a full packet capture of all packets, decrypt them all, and keep the trace running all the time. This kind of trace is necessary because analysts don't know which web or app server, for example, is going to be hit for a particular user session, and they don't know when an intermittent problem is going to hit. Instead, enterprises have built up robust, out-of-band packet sniffing devices with TLS decryption capability fed by passive network taps and/or passive mirror ports.

Endpoint analysis also misses the crucial troubleshooting function of isolating the fault domain of a problem among many infrastructure devices between the TLS endpoints.

3.7 Security Monitoring at the Endpoint

Network security monitoring is done by a number of purpose built network devices such as IDS/IPS and security analysis solutions. Network based fraud detection applications can include multiple servers and databases that all communicate with each other. It's not feasible to put all this functionality into an endpoint and have its normal workload unaffected.

Endpoints can be overwhelmed by too much security monitoring and their performance impacted. Networks can also be overwhelmed by extensive security reporting from endpoints. As a result, endpoint monitoring is often scaled back to a level that the endpoint and its network connection can handle.

Endpoints cannot be completely trusted for network security monitoring. Malware can delete logs and turn off future logging. It's also not always possible to secure data stored on an endpoint,

for example, if the endpoint is a laptop and the user packs it up and

walks out of the enterprise. The great variety of endpoint types also makes it difficult to implement a consistent monitoring strategy using endpoints alone.

Network security monitoring is an important complement to endpoint security monitoring, and is part of a "defense in depth" strategy.

3.8 Encrypted Traffic Inspection

This technology, while interesting and applicable in some situations, does not fully satisfy the requirements of enterprise traffic inspection.

From an application performance and availability perspective, encrypted traffic inspection will not figure out severity one slowdowns or outages, or any other level of problem that may hit an enterprise. Large enterprises have thousands of unique applications that all behave differently at layer 7, and any one of these applications may need layer 7 analysis when a problem hits. This factor of troubleshooting alone is enough to make encrypted traffic inspection an unacceptable, or at least incomplete, solution for enterprise encryption problems.

Encrypted traffic inspection does not address fraud detection for either internal or external fraud, both of which look at decrypted TLS packets.

Encrypted packet inspection does not address Application Performance Monitoring, Customer Experience Monitoring, or the use of decrypted packets for regulatory compliance monitoring.

From a security perspective, encrypted traffic inspection is not going to detect every zero day attack. The parameters it is looking for in the TLS handshake can be varied by new malware. Encrypted traffic inspection, for some methodologies, may be less effective under TLS 1.3 when the handshake is encrypted.

Encrypted traffic inspection doesn't take into account the manual, deep packet inspection done by threat detection teams in order to analyze malware alerts, track down their source, and to identify if

an attack succeeded or failed.

3.9 IPsec instead of TLS

The enterprise rollout of internal TLS has been a multi-year project. Enterprises can't just flip a switch and start running IPsec. Moving to IPsec would likely be a multi-year and expensive project. There is extensive manual configuration that would need to be done.

There are a number of infrastructure services that don't support IPsec today. For example, IPsec is not supported today by load balancers for data center load balancing services. IPsec is also not supported by Internet proxies for outbound Internet services.

A number of new IETF protocols are tied to TLS, including HTTP/2, DPRIVE, and QUIC. Enterprises need a TLS decryption solution in order to support these protocols.

4. An Examination of Arguments Against All Network Decryption

4.1 Technical Arguments

4.1.1 "I work for a large company and we don't have to decrypt packets."

Large Internet companies being put forward as examples of promoting encryption are not necessarily encrypted through their entire private enterprise environment as some financial and health care institutions are.

There are varying levels of data center depth and complexity between enterprises. Some enterprises have a flatter data center structure, depending on the kinds of services they offer. Other enterprises have many layers to their applications, with multiple layers of infrastructure like firewalls and load balancers, and many layers of middleware, authentication, fraud detection, mainframe, etc. There are also legacy applications that add complexity to the infrastructure, and add to the requirement for decrypted packet payload analysis.

Network packet decryption, if it is happening, is likely not visible to all employees within an enterprise. Typically, only select groups

within an organization utilize or are aware of this level of detail.

As more enterprises in different verticals add TLS decryption inside their data centers, they are going to realize that they also have a need for out-of-band TLS decryption.

4.2 Privacy Arguments

4.2.1 "It's a violation of personal privacy to decrypt TLS traffic anywhere except at the TLS endpoint."

Enterprises have many legitimate business reasons for inspecting their own data, and the IETF should provide them with well studied and standardized options that meet these critical business needs.

All TLS data is already being decrypted multiple times in the enterprise data center, and customer data is already available to certain employees of that enterprise. Network packet decryption is just one more decryption of its own data by the same enterprise.

4.2.2 "Pervasive Monitoring is an Attack"

Pervasive monitoring inside the enterprise has many legitimate use cases, including troubleshooting and network security monitoring. As an example, some applications are too complicated to troubleshoot at the network packet level without advanced preparation of the packet level monitoring, meaning that it takes too much time during a critical outage to get traces set up in all the right places. The answer to this is pervasive monitoring of that application or that environment so that network packet traces, including TLS decryption, are ready when a problem hits.

RFC 7258 [RFC7258] should be modified to account for the many enterprise use cases where pervasive monitoring is not an attack.

5. Possible TLS 1.3 Decryption Solutions

5.1 Static Diffie Hellman

Static Diffie Hellman [draft-green] as described in draft-green-tls-static-dh-in-tls13 meets the enterprise need in a manner similar to

running RSA key exchanges and using static RSA private keys. Enterprises would be obligated to protect their static keys as they are today in the RSA environment. Draft-green requires no changes to the TLS client, and no changes to the TLS 1.3 spec. It has no impact on the CPU load of the TLS server. Enterprises have the option of rotating their static Diffie-Hellman private keys as often as they see fit.

5.2 TLS 1.3 Option for Negotiation of Visibility in the Datacenter

TLS 1.3 Option for Negotiation of Visibility in the Datacenter [draft-rhrd] as a solution has some alternative features in comparison to draft-green [draft-green]. It eliminates static Diffie-Hellman private keys from the TLS server as in the case of draft-green. The key manager would only write static private keys from the SSWrapDH1 key pair to the decryption appliances in the protected enterprise network. Draft-rhrd provides for client opt-in and visibility on the wire that traffic payload inspection may be happening. It also allows for decryption in the case of session reuse, which solves a large problem for enterprise monitoring and troubleshooting. It will have some impact on the CPU load of the TLS server.

5.3 Solution Summary

For both of these solutions, a standard is needed so that all the related systems will interoperate. Draft-green [draft-green] would need to be implemented by multiple TLS server vendors, multiple decryption appliance vendors, and multiple key management solutions. Draft-rhrd [draft-rhrd] would require all of these in addition to implementation by TLS clients. For the above reasons, custom code is a highly unattractive, and possibly unworkable solution.

6. Conclusion

Out-of-band TLS decryption is used by a number of enterprise tools for mission critical functions, and it supplies ubiquitous packet payload visibility that can't be replaced by other methods. Endpoint analysis is limited by its lack of robustness for analytic activities, by the fact that it can't be completely trusted, and by its blindness to issues in the intervening infrastructure. Inline/MITM decryption adds cost, latency, and production risk at

every point it is implemented, and it doesn't scale to meet the requirements of the use cases presented above.

7. Security Considerations

There are security tradeoffs that enterprises should be allowed to decide. On the one side are the benefits of Forward Secrecy inside the enterprise. On the other side are the benefits of ubiquitous packet payload visibility inside the enterprise. Enterprises are most qualified to make this business decision for themselves, and the TLS Working Group should provide them options rather than making the decision for them.

Enterprises choosing to do out-of-band decryption need to continue to implement whatever security controls are appropriate for protection of this decryption environment, including protection of keys, controlling access to the decrypted data, etc.

8. IANA Considerations

There are no IANA considerations.

9. References

9.1 Normative References

[draft-green] Green, M., Droms, R., Housley, R., Turner, P., and S. Fenter, "Data Center use of Static Diffie-Hellman in TLS 1.3", draft-green-tls-static-dh-in-tls13-01 (work in progress), July 2017.

[draft-rhrd] Housley, R. and Droms, R., "TLS 1.3 Option for Negotiation of Visibility in the Datacenter", draft-rhrd-tls-tls13-visibility-01 (work in progress), March 2018.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[RFC7258] Farrell, S. and Tschofenig, H., "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <https://www.rfc-editor.org/info/rfc7258>.

[TLS13] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", draft-ietf-tls-tls13-26 (work in progress), March 2018.

9.2 Informative References

Acknowledgments

Nalini Elkins, Mike Ackermann, Darin Pettis, and Russ Housley contributed through discussion to the development of this document.

Authors' Addresses

Steve Fenter
Enterprise Data Center Operators, Inc.
36A Upper Circle
Carmel Valley, CA 93924

EMail: info@e-dco.com