Looking Under the Rock: Deployment Strategies for TLS Decryption

GIAC (GCIA) Gold Certification

Author: Chris Farrell, chrisf@iquest.net
Advisor: Stephen Northcutt
Accepted: December 29, 2017

Abstract

Attackers can freely exfiltrate confidential information all while under the guise of ordinary web traffic. A remedy for businesses concerned about these risks is to decrypt the communication to inspect the traffic, then block it if it presents a risk to the organization. However, these solutions can be challenging to implement. Existing infrastructure, privacy and legal concerns, latency, and differing monitoring tool requirements are a few of the obstacles facing organizations wishing to monitor encrypted traffic. TLS decryption projects can be successful with proper scope definition, an understanding of the architectural challenges presented by decryption, and the options available for overcoming those obstacles.
1. Introduction and TLS Use Cases

The internet has evolved from its beginnings as a simple connection between universities to the vast global network we know today. It acts as the information infrastructure which supports many aspects of our lives. Critical confidential information travels across the internet continually. The original developers of the internet focused on transferring information that recipients would understand upon arrival (Gromov, 1995). Protecting confidential data in transit would later be addressed using encryption which utilizes Transport Layer Security (TLS).

Web communications

Web browsing is the most prevalently used internet application. An estimated 4.6 billion web pages exist on the internet; that number continues to grow steadily (Kunder, 2017) The servers of the original World Wide Web were designed to produce static web pages quickly. The information created was considered public information functioning similarly to a roadside billboard.

Over time, the need to deliver information only to intended recipients evolved. A method had to be designed to protect the data from being intercepted by unintended audiences. Jinwoo Hwang describes in detail the historical evolution of SSL to TLS v1.3 in his research at the IBM DeveloperWorks library (Hwang, 2012) (Whitteker, 2016)

TLS-VPNs

TLS VPNs are another popular use of TLS technology. Instead of encrypting only the web traffic to/from a web server, the TLS VPN encapsulates all sensitive traffic. This encryption protects all applications with TLS encryption. At the destination, the endpoint server decrypts and delivers the data.

An example is a remote employee who needs to connect to the employer's network. Other encryption technologies exist which provide this functionality, but their use typically requires firewall configuration changes at each site the employee may visit. However, TLS outbound traffic is enabled on most firewalls to allow secure web browsing. A TLS-VPN connection takes advantage of this already-open pinhole through

Chris Farrell, chrisf@iquest.net
the outbound firewall to create the encrypted connection (Ferrigni, 2003). The ability to encrypt almost any communication within a TLS-VPN tunnel is powerful. It protects the encrypted traffic from any prying eyes on the networks which the traffic traverses. This capability is also useful to those who wish to move malicious traffic across networks. The encryption protects the attacker’s traffic from being monitored and detected by the network’s defenders.

2. Malicious Use Cases

The obstruction created between those who can decrypt the communication and those who cannot decrypt allows commercial transactions and private information to be shared securely. However, the obstacle which protects sensitive, legitimate traffic from unauthorized access also can be used by those with nefarious intentions to hide their activities from security monitoring systems. Any attacker with the goal of communicating from within a company to any external server now has a very convenient way to make the connection and hide the details of the communication within the encrypted connection.

Command and Control

Malware developers create software for delivery to computers within secured network perimeters. Once the malware has been successfully delivered and executes on an internal resource, the next step is to make an outbound connection across the secured perimeter to a Command and Control (C&C) server to receive further instructions. Security teams monitor for these types of outbound communications and attempt to detect and block unauthorized traffic (Butler, 2013). Through the use of TLS, attackers can send these outbound communications undetected to servers on the internet leveraging the openings that have been created to allow legitimate web communications. (Liburdi, 2017)
**Exfiltration**

Once the malware has been established on an internal device, the next steps are privilege escalation, lateral movement, and persistence. This process almost always is done to access the sensitive internal information that the network perimeter was intended to protect.

In most cases, the adversary’s goal is to exfiltrate the data outside the secured network perimeter. If the attacker utilizes a TLS tunnel similar to a TLS-VPN, the monitoring systems will be unable to detect the data hidden within the encrypted session. (Antwerp, 2011) What’s needed is a solution which exposes the hidden data to the enterprise while maintaining the TLS connection across the internet.

### 3. Network-Based Decryption

TLS encryption introduces a security blind spot because the data cannot be read and evaluated. The security monitoring systems must be able to assess the data in its unencrypted state. The security monitoring systems can then send alerts or block it as needed.

Network-based TLS decryption involves placing a gateway server between all of the devices within the secured perimeter and the internet. The internal devices are configured to recognize and accept the gateway as an intermediary for all outbound TLS connections. When an internal device initiates a legitimate TLS connection, that device attempts to connect to the external destination and the gateway intercepts the connection.

By acting as a man-in-the-middle, the gateway pretends to be the external destination and creates an encrypted session with the internal device. This connection allows the gateway server to decrypt the data for inspection. If done correctly, persons using the web browsers are not aware the traffic is flowing through an interceptor. The result is that the browser reports a secure connection with the external destination exists.

Chris Farrell, chrisf@iquest.net
After the interceptor has decrypted the traffic, the clear-text data can be evaluated by security monitoring systems for detection of malicious activity. The security monitoring systems deliver alerts or terminate the TLS session, based on the determination. If the data is determined to be legitimate, the gateway server then creates a separate encrypted session with the external destination. The destination server does not know it is communicating with a proxy for the device within the secured perimeter.

4. Decryption Obstacles and Solutions

Organizations should consider the obstacles presented by legal and privacy concerns, architectures, modes of operation, certificates, latency, support and high availability when deploying a technology such as TLS interception.

Legal Considerations

TLS is one of the most trusted encrypted transports for the international computing community. With so many people using it to protect their communications, and internet cloud technologies continuing to expand across geographic borders, it is essential to consider how international data privacy laws apply to these activities.

Individual countries enact data privacy laws with differing degrees of maturity and complexity. In some instances, communities may work in partnership to create legislation. This form of cooperation is most evident in the European Union (EU) where 28 countries have adopted uniform, more stringent data privacy laws.

The EU implemented Data Protection Directive 95/46/EC in March 2000. EU countries are scheduled to adopt the General Data Protection Regulation on May 25th of 2018 (Data Protection Laws of the World, 2017). Although this is an EU regulation, it applies to any organization regardless of their physical location if they are collecting personal data of EU residents. The law applies to the residence of the data owner, not the geographic location of the interception. Another important aspect of this law is the concept of data ownership. The subject of the data owns the personal information. The subject must be able to withdraw their consent easily. Additionally, owners have the
"Right To Be Forgotten" and must be given a process to have any personal data purged from any monitoring systems.

If a company intends to intercept TLS traffic for information security monitoring, it must address several fundamental tenets of the law. The first is informed consent which means that before the company can intercept TLS, it must inform everyone whose data it will monitor. The enterprise also must inform the data owners about why it needs the data and how it will use the data. Data owners must explicitly grant written consent. The company must establish a process to allow the data owner to withdraw consent at any time. Companies may make this a requirement of employment, but careful security architecture must be applied so that it will monitor only those who have given consent to TLS interception.

Another important aspect of the law is that for any data retained about an individual, a process must be established to allow the individual access to the data free of charge and be able to confirm how the enterprise used the data. Because the value of intercepted traffic for security reasons diminishes quickly over time, a policy of deleting the data shortly after it is no longer useful reduces the burden of this requirement. This practice also addresses several other requirements such as the right to correct inaccurate data, the "Right-To-Be-Forgotten" and data portability. Accordingly, the company must also be able to demonstrate compliance with these requirements through proper documentation, logging, and continuous risk assessments (Preparing for the General Data Protection Regulation, 2017).

**Privacy Concerns**

In countries where the applicable laws are not as well defined, privacy is still a concern. With the rapid advancement of technology, it has been difficult for the legal community to craft laws which address the employee’s expectation of privacy (Smith, 2012). Many factors can contribute to a common law claim to a reasonable expectation of privacy when a company implements TLS interception. An example is when the company does not own the device being used to encrypt the communication. If an employee uses a personal device on the company's WiFi hotspot, the employee's expectation of privacy may be objectively reasonable. This scenario is especially

Chris Farrell, chrisf@iquest.net
pertinent if the policies do not explicitly address personal device communications on the company network. It may be preferable to exclude the WiFi hotspot from TLS interception altogether.

It also may be desirable to exclude some communications from TLS interception based on the data itself. Examples include communications to finance, banking, wellness, and healthcare sites. If an employer intercepts a TLS session between an employee and his or hers doctor’s website, the employer should be prepared to protect the Private Health Information (PHI) to the standards established by the Health Insurance Portability and Accountability Act (HIPAA) (Department of Health and Human Services, 2000). This example and others related to their respective industries need to be weighed carefully for the mutual protection of both the employer and the employee.

**Serial vs. Parallel Deployment**

When considering TLS interception, most companies wish to apply key information security technologies to the encrypted traffic. There are many of these technologies, but three common applications are; Web Proxy, IDS/IPS Signature Inspection, and Data Loss Protection. Web proxies monitor which websites users visit and what data users send to these websites. IDS/IPS detects malicious traffic based on known traffic signatures. Data Loss Prevention detects the presence of confidential information leaving the secured company perimeter.

### 4.1 Multiple Interception with Serial Distribution

There are several approaches to providing a decrypted data stream to these applications. A straightforward approach would be to align the technologies in series and allow each device to decrypt the traffic individually as depicted in Figure 1 – Multiple Interception with Serial Distribution below.

![Figure 1 – Multiple Interception with Serial Distribution](image-url)

Chris Farrell, chrisf@iquest.net
The primary issues with this approach are the delays introduced as each device decrypts/re-encrypts the data and the increased risk of an interruption of service if any of these devices encounters an error. Latency grows every time the decryption process occurs. Intercepting the data once can be noticed by users. If the traffic must be intercepted three times before it reaches its destination, the user may have a negative experience. Additionally, the architecture presented in Figure 1 – Multiple Interception with Serial Distribution requires additional expenses to provide the resources needed to support the interception on each device (Shackleford, 2012). It quickly becomes apparent one should avoid this approach because of the technical and financial ramifications.

Multiple Interception with Serial Distribution configuration does allow the company to continue to use existing security monitoring systems without replacement. However, the security monitoring systems would likely need additional resources to support TLS interception functionality.

4.2 Single Interception with Serial Distribution

A dedicated decryption gateway can be used to offload the interception tasks to reduce the resources and latency involved with intercepting the outbound encrypted stream multiple times. In the Figure 2 – Single Interception with Serial Distribution configuration, the decryption gateway accepts and terminates the TLS session from the originating internal device. It then converts the data to an unencrypted clear text stream and delivers it to all security monitoring systems before returning the data to the dedicated decryption gateway. The process continues with the gateway establishing the outbound encrypted session with the final destination.

Chris Farrell, chrisf@iquest.net
Because the TLS interception occurs only once, the interception latency is introduced to the stream once. Because a single device handles the load, this approach reduces resource requirements and costs.

It is important to note once the TLS interceptor passes the decrypted clear-text stream to the first security monitoring device, the data travels unprotected across the network. The network segments which carry the clear-text data must be physically segmented, not logically segmented, from all other traffic. Anyone in control of a network interface within the same network where this clear-text data is present can easily extract and compromise it.

The path through the decryption device and the security monitoring systems is still a serial path. As a result, if any of the security monitoring systems were to fail, it would result in a disruption of service. This approach allows utilization of the existing security monitoring systems to continue without replacement. Because these systems are not involved with the interception responsibilities, the systems do not need additional resources for these purposes.

4.3 Single Interception with Parallel Distribution

To remove the possibility of any individual security system disrupting the communication between the user and the external server, the concept of a dedicated
interception device is again considered. Instead of sending a single stream of data out to a series of security monitoring systems, the decrypted stream is duplicated and sent to each monitoring system individually as depicted in Figure 3 – Single Interception with Parallel Distribution. This parallel deployment strategy maintains the single interception point of the TLS data while removing the threat of any monitoring system disrupting the data stream as seen in the Single Interception with Serial Distribution model.

![Figure 3 – Single Interception with Parallel Distribution](image)

Similar to the Single Interception with Serial Distribution model, the potential for the clear-text data to be retrieved from the network still applies. The network segments which carry the clear-text data must be physically segmented, not logically segmented, from all other traffic.

It also is important to note how a parallel deployment introduces the ability to use encapsulation protocols such as the Internet Content Adaption Protocol (ICAP) which encapsulates the content between the TLS interceptor and the security monitoring device (Elson, 2003). This protocol is commonly used to inspect web traffic for data loss prevention activities. The advantage of this option over a clear-text stream is that the security monitoring system is not required to return the data to the TLS interceptor. Instead, it returns a response indicating whether the data sent should be allowed to continue to its ultimate destination.

Chris Farrell, chrisf@iqest.net
This parallel deployment introduces a new obstacle which can be challenging to resolve should it arise. If the TLS interceptor is configured in blocking mode (which is referenced in the section, Blocking vs. Detection Modes), it must wait until each security monitoring system has completed evaluating the stream before re-encryption can occur. This delay in processing may require an upgrade of a security monitoring system so it can respond as quickly as the other systems. Diagnosis and remediation of this issue can be difficult at times, especially if different vendors are involved.

This approach allows existing security monitoring systems to continue without replacement. Because these systems are not involved with the interception responsibilities, additional system resources are not needed.

### 4.4 Single Device Deployment

The final approach involves integrating all the security monitoring functions into one device to remove the risk of exposing the clear-text data onto networking equipment where it could be extracted for nefarious purposes. Again, this supports the concept of a single interception of the encrypted session between the user and the external device which reduces the latency. Instead of using separate, security monitoring systems, these systems integrate directly into the TLS interception appliance. These features typically are offered in “Unified Threat Management” or “Next Generation” firewalls such as the FortiNet FortiGate brand.

Additional benefits to using this approach are the increases in speed achieved by maintaining the clear-text data within a single chassis and the integration of the services provided. Instead of using multiple vendors and attempting to understand why the devices from different companies are not cooperating, a single vendor would render support which would accelerate troubleshooting. In Figure 4 – Single Device Deployment, all TLS interception and security monitoring activities occur within the Unified Threat Management device.
Blocking vs. Detection Modes

Security monitoring systems can be broadly categorized into detection or prevention technologies. The goal of detection technologies is to monitor for specific patterns of data or behavioral characteristics. Once a match is detected, alerts are sent to personnel to investigate further. Preventative technologies must also detect the patterns or characteristics, but they take the additional step to block the communication as it is occurring. Blocking the communication reduces the containment time for attacks and in the case of data loss prevention, can prevent an incident altogether. It is important to decide whether the security monitoring system which receives the clear-text data provided by the TLS interception will be expected to stop the attempted communication or detect it (Holland, 2004).

If detection is all that is required, the security monitoring system can receive a clear-text copy of the transmission and alert out-of-band from the TLS interception activities. The results of the monitoring system have no bearing on whether the TLS data
will be re-encrypted. This process removes many of the issues associated with latency and timing because the TLS interceptor does not have to wait for an acknowledgment from the security monitoring devices before proceeding with the re-encryption activities.

Blocking mode requires one of these two strategies: In-line deployment or interceptor block recognition. In-line deployment places the security monitoring system as a required participant in the routing of the TLS traffic. As such, if the system detects data which should be blocked, it simply stops the transmission itself, and no further action is needed to stop the communication. This approach can be seen in Figure 1 – Multiple Interception with Serial Distribution and also in Figure 2 – Single Interception with Serial Distribution.

Interceptor block recognition requires the TLS interceptor to establish a method of communicating with the security monitoring systems so that the interceptor is notified of the determination reached by the clear-text data analysis. If the communication is to be blocked, the interceptor must not re-encrypt the data and ideally will relate the result back to the user (Elson, 2003). This process can be seen in Figure 3 – Single Interception with Parallel Distribution.

The Single Interception with Parallel Distribution model introduces the timing issue mentioned in the description of that model. The TLS interceptor must receive a positive acknowledgment from every security monitoring system configured to be in blocking mode before it can proceed with the re-encryption of the data for external delivery. Those who configure the interceptor must decide what actions will be taken if no response is received from a security monitoring system. If the decision is made to allow the re-encryption to occur without an acknowledgment from the system, the enterprise is essentially accepting the risk associated with the outbound data. If the decision is made to block traffic whenever an acknowledgment is not received, it could negatively impact the user experience and create downtime.

The Single Device Deployment model carries the same concerns for blocking as the Single Interception with Parallel Distribution model with the difference being that the communication is occurring between different processes within a single device. This integration promises to reduce the conflicts introduced by attempting to use multiple

Chris Farrell, chrisf@iquest.net
vendors to perform these tasks. Vendor conflicts are removed, the clear text traffic remains contained within the device, and communication occurs at higher rates. If existing security monitoring devices aren’t being integrated, Single Device Deployment is favorable.

Certificates

TLS sessions can be established for secured communication between two systems for a variety of applications. The most widely used application is between a web browser and a web server. These connections are established by the following means:

- Browser contacts web server.
- Server sends a copy of its certificate which includes the public key
- Browser checks the certificate against its list of trusted certificate authorities and checks the validity of the certificate
- If the server certificate is trusted, the browser creates a symmetric encryption key for the TLS session and encrypts the symmetric session key with the server’s public key. The browser sends this back to the server.
- The server decrypts the symmetric session key with its private key and sends an acknowledgment back to the browser using the symmetrical encryption key for the first time. All subsequent communications are encrypted with the symmetric session key.

The first obstacle to overcome is to convince the browser that the certificate provided by the TLS interceptor should be accepted (Dormann, 2015). When one adds the interceptor's certificate authority (CA) to the browser's root CA store, it accomplishes this task. This can typically be automated with an Active Directory Group Policy Object to push the certificate into the "Trusted Root Certification Authorities" store on the workstations.

Chris Farrell, chrisf@iquest.net
Latency

TLS interception processing introduces latency. The best user experience will always be without TLS interception for monitoring. Each operation undertaken by the interceptor creates delays into the communication between the internal client and the external server. It is significant that these delays only be applied once. One should avoid the use of multiple decryption/re-encryption cycles, like those in the model which uses Multiple Interception with Serial Distribution (Pirc, 2013).

When utilizing one of the Single Interception models, latency from the decryption/re-encryption cycle is minimal, but a different latency obstacle may present itself. This impediment is the latency introduced by the clear-text traffic traversing the network. Each hop between the interceptor and the security monitoring systems adds a delay. The delay is comparably less than the decrypt/re-encrypt operation, but the multiple hops between the devices can escalate quickly to create a dissatisfied user. To reduce these network delays, the data moving between the interceptor and the security monitoring systems should occur on a physically-dedicated, high-speed network.

Latency in the Single Interception with Serial Distribution model would have the network latency introduced every time the clear-text data moves from one security monitoring device to the next in the serial configuration. However, once the traffic has completed the circuit back to the interceptor, the interceptor can immediately begin re-encrypting the data for delivery to the external destination.

This scenario is not valid for the Single Interception with Parallel Distribution model. In this case, the interceptor must create separate streams to deliver to each security monitoring system. Then, it must wait for an acknowledgment from each before the re-encryption process can occur. In the case of clear-text streaming, the clear text returning to the device is the acknowledgment. For ICAP streaming, the interceptor waits for the positive reply. Either way, the interceptor may be forced to "time-out" while waiting for a response.

Chris Farrell, chrisf@iquest.net
Support and Interoperability

Except for the Single Device Deployment model, in which all of the security monitoring systems are contained within one device, maintaining a TLS interception solution will require support from all of the vendors involved. Many vendors will have an offering of their own for their proprietary product to intercept TLS. Working with a different interception solution may be difficult for them to support. If a stubborn problem arises, it may require additional effort to bring everyone together to find a solution. It is important to verify all vendors involved can and will support the chosen interception model.

High Availability

High availability has not been addressed to simplify the explanations of the individual architectures. Whichever architectural model is chosen to support TLS interception, it must be deployed redundantly to prevent downtime. Evaluations should be done to assess opportunities to reduce the risk of a disruption of service.

Solutions can become cost prohibitive for serially distributed architectures where each security monitoring system would need duplication. In the parallel deployments, the interceptor may be configured to allow the traffic to pass if a security monitoring system produces no response. This configuration may be an acceptable risk while troubleshooting.

Chris Farrell, chrisf@iquest.net
5. Conclusion

Before launching a project that involves TLS interception, it is necessary to consult with legal counsel to fully understand the implications of exposing protected personal information. With the number of remote and mobile employees growing, so do the chances that the laws of foreign countries will apply to the data.

Privacy should be considered to determine the employee’s expectation of privacy and the responsibilities incurred when the employer comes into possession of the employee’s personal information. The regulatory requirements affecting the security of the decrypted data may introduce additional requirements for a TLS implementation.

The architecture chosen to deploy the solution must take into consideration several essential concepts. The first is to decide which architectural model integrates best with the existing security monitoring systems and network topology. Efforts should be made to avoid TLS interception more than once during any given communication. Anything more than one can negatively impact the user experience and will create unnecessary complexity. The distribution of the clear-text data should occur on a dedicated physical network to reduce delays and to allow for strict security controls.

Decisions also will need to be made about how the security monitoring systems should react if any nefarious activity is detected. If blocking actions are required, it will introduce complexity. If detection is the only requirement, parallel distribution can reduce complexity and latency.

The internal client systems which will use TLS passing through the interceptor must trust the certificate presented. This trust is established by inserting the root certificate authority for the interceptor's certificate into the client's certificate store for trusted certificate authorities.

Lastly, one should proactively establish support arrangements with all vendors involved in the project to gain buy-in and allow for smooth troubleshooting for any issues that may arise.

Chris Farrell, chrisf@iquest.net
In the Addendum can be found lab testing results which compares the two most efficient SSL interception architecture strategies. The Single Device Deployment architecture was chosen to represent a deployment without existing security monitoring devices. Single Interception with Parallel Distribution was selected to represent inserting TLS interception when leveraging existing security monitoring systems for detection purposes.

This research focused on network-based TLS interception to provide insights into the obstacles companies face when deploying this technology. An alternative is to deploy software agents directly to all workstations within the secured perimeter. Further research into the obstacles and benefits of agent-based TLS interception would also be beneficial.

Chris Farrell, chrisf@iquest.net
6. References


Excellent resource for international data privacy law comparisons


Chris Farrell, chrisf@iquest.net


Chris Farrell, chrisf@iquest.net

© 2018 The SANS Institute Author retains full rights.


Looking under the Rock

November 23, 2017, from


Chris Farrell, chrisf@iquest.net


Chris Farrell, chrisf@iquest.net
7. Addendum

A test lab was constructed to compare the two most efficient SSL interception architecture strategies. The Single Device Deployment architecture was chosen to represent a deployment without existing security monitoring devices. Single Interception with Parallel Distribution was selected to represent inserting TLS interception when leveraging existing security monitoring systems.

Testing Process

The web server was configured with a simple HTML form and a PHP script to transfer files from the browser’s local file system to the web server. Sample files were created to test detection after decryption. One large data file containing random data was created to test throughput.

A certificate was purchased from a public certificate authority for the web server. This certificate was used to encrypt the communication between the browser and the web server using TLS v1.2.

Phase 1 of the lab testing was a control experiment. The FortiWiFi was configured to mimic the capabilities of a simple router between the browser and the web server. It also acted as a firewall and network address translation device for internet access. This fundamental architecture would be the foundation to compare latency and throughput in later experiments.

In Phase 2 of the testing, the FortiWiFi was configured to act as a TLS interceptor in the Single Device Deployment architecture. The root certificate was imported into the certificate store of the web browser which allowed the FortiWiFi to intercept the traffic without triggering errors. Testing was repeated as each additional security service was added and the results recorded.

Finally, in Phase 3, the FortiWiFi was configured to intercept the TLS communication and deliver the clear-text data to external security monitoring systems for detection. This architecture represented the Single Interception with Parallel Distribution.
architecture. Testing was repeated as each additional security service was added and the results recorded in the tables below.

The data revealed the fastest transfer times occurred within the Single Device Deployment architecture. Surprisingly, when the firewall was performing IPS monitoring alone, performance degraded significantly. As long as the DLP or antivirus modules were being utilized, the performance improved. IPS activities alone accounted for the longest file transfer time in this configuration.

In the Single Interception with Parallel Distribution architecture, the performance was almost identical to the results seen in the aforementioned IPS-alone scenario. The times increased slightly for each additional interface being supplied with a clear-text feed.

In all scenarios, the load introduced onto the firewall by TLS interception was substantial. The FortiWifi Model 60D is an entry-level, next-generation firewall. In this isolated environment with only one user, the processor hovered near 90% utilization anytime a large file was transferred. The time differentials reflect this load and the impact a user would recognize. Most firewall manufacturers utilize hardware accelerators in more expensive models to counter the TLS interception processing load.
Lab Testing Results

<table>
<thead>
<tr>
<th>Single Device Deployment Architecture</th>
<th>97MB Random Data File Transfer Time</th>
<th>Internal Detections</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Interception</td>
<td>7s</td>
<td>None</td>
</tr>
<tr>
<td>IPS Monitoring Only</td>
<td>1m33s</td>
<td>Malware*</td>
</tr>
<tr>
<td>Virus Monitoring Only</td>
<td>48s</td>
<td>Virus</td>
</tr>
<tr>
<td>DLP Monitoring Only</td>
<td>47s</td>
<td>CCNs/SSNs</td>
</tr>
<tr>
<td>DLP Monitoring, Virus Monitoring, IPS Monitoring</td>
<td>47s</td>
<td>Virus CCNs/SSNs Malware</td>
</tr>
</tbody>
</table>

*The large file transfer times increased substantially. The malware is blocked but no notification is presented to the browser. Issues are not present when additional monitoring is done in conjunction with IPS monitoring.*

<table>
<thead>
<tr>
<th>Single Interception with Parallel Distribution Architecture</th>
<th>97MB Random Data File Transfer Time</th>
<th>External Detections</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Interception</td>
<td>7s</td>
<td>None</td>
</tr>
<tr>
<td>Clear-Text to One Interface for IDS</td>
<td>1m20s</td>
<td>Malware</td>
</tr>
<tr>
<td>Clear-Text to Two Interfaces for IDS / DLP</td>
<td>1m23s</td>
<td>Malware CCNs/SSNs</td>
</tr>
<tr>
<td>Clear-Text to Three Interfaces for DLP / Virus / IDS</td>
<td>1m36s</td>
<td>Malware CCNs/SSNs</td>
</tr>
</tbody>
</table>

Chris Farrell, chrisf@iquest.net
Lab Diagram

- Internet
  - DNS Server
  - CA Server

Virtual Server
- Web Server
- IDS/IPS
- Web Proxy
- DLP

SSL Interceptor
- Browser

Data Files

Five files were used to test speed and compare detection capabilities:

1.) EICAR Virus Signature File
2.) Social Security Number File
3.) Credit Card Number File
4.) Malware Simulation File (mimics Hydraq Trojan C&C traffic)
5.) A 97MB file of random base64 encoded data

Chris Farrell, chrisf@iquest.net
## Lab Equipment

| Client | AMD FX-8370 8-core processor  
|        | 24GB RAM  
|        | Gigabit Network Adapter  
|        | Ubuntu Linux 17.04 w/ Chrome Browser V62.0.3202.94 |
| Firewall | Fortinet FortiWiFi 60D UTM Firewall  
|          | Firmware v5.6.2  
|          | Gigabit Network Adapters |
| Network | Nortel POE Gigabit Switch |
| Virtual Server | HP Compaq 8200 Elite  
|                | Intel i5-2400 4-Core Processor  
|                | 22GB RAM  
|                | Gigabit Network Adapter  
|                | VMware ESXi 6.0.0 |
| Web Server (VM) | 2 vCPUs  
|                 | 4GB RAM  
|                 | Ubuntu 16.04 w/ Apache 2.4.18 (Ubuntu) |
| IDS/IPS Server (VM) | 2 vCPUs  
|                   | 4GB  
|                   | Ubuntu Linux 15.04 w/ Snort V2.9.9.0 |
| Web Proxy Server (VM) | 2vCPUs  
|                       | 4GB  
|                       | Ubuntu 16.04 w/ Squid V3.5.12 |
| DLP Server (VM) | 2vCPUs  
|                  | 2GB  
|                  | Ubuntu 15.04 w/ Squid V2.9.9.0 & PigPen |

Chris Farrell, chrisf@iquest.net
LAMP Web Server Modifications

Commercial certificate was applied to the web server to allow the test server to properly represent an external server while being able to maintain consistent communications for testing.

Web server settings are adjusted to allow large file transfers as well by editing the "php.ini" file:

memory_limit = 1280M  
post_max_size = 100M  
upload_max_filesize = 100M (this is the maximum size)
Very simple web pages are added to the server to facilitate file uploads:

index.html

```html
<!DOCTYPE html>
<html>
<body>
<form action="upload.php" method="post" enctype="multipart/form-data">
Select image to upload:
<input type="file" name="fileToUpload" id="fileToUpload">
<input type="submit" value="Upload Image" name="submit">
</form>
</body>
</html>
```

upload.php

```php
<?php
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]['name']);
$uploadOk = 1;
$imageFileType = pathinfo($target_file,PATHINFO_EXTENSION); $check = getimagesize($_FILES["fileToUpload"]['tmp_name']); // Check if $uploadOk is set to 0 by an error
if ($uploadOk == 0) {
    echo "Sorry, your file was not uploaded.";
    // if everything is ok, try to upload file
} else {
    if (move_uploaded_file($_FILES["fileToUpload"]['tmp_name'], $target_file)) {
        echo "The file ". basename( $_FILES["fileToUpload"]['name']). " has been uploaded.";
    } else {
        echo "Sorry, there was an error uploading your file.";
    }
}
?>
```

Chris Farrell, chrisf@iquest.net
Browser Modifications

Initial connections to the web server worked without issue:

![Image of browser connection]

After SSL is intercepted, the browser presents an error:

![Image of browser error]

The Fortinet root CA was imported into the trusted CA store to resolve the issue.

---

Chris Farrell, chrisf@iquest.net
While using the Single Device Deployment architecture the firewall produced warning pages for most test data:

![FortiGuard IPS & Application Control](image)

**IPS Sensor Triggered!**

Your attempt to access the internet resource is blocked by IPS Sensor.

- **URL:** https://sslintercepttarget.rapidfault.com/upload.php
- **Client IP:** 172.16.1.23
- **Server IP:** 208.88.250.45
- **User name:**
- **Group name:**
- **Policy:**
- **FortiGate Hostname:** FWF60D

In Single Interception with Parallel Distribution model, the TLS session between the browser and the web server was intercepted and the unencrypted data was delivered to the designated firewall interfaces:

**Firewall Config Snippet:**
```
config firewall policy
edit 57
set name "PrivatetoLAMP"
set uuid 13a22fa8-d06d-51e7-b7ad-330e6f2e489a
set srcintf "Private"
set dstintf "virtualDMZ"
set srcaddr "all"
set dstaddr "LAMP Server"
set action accept
set schedule "always"
set service "ALL"
set utm-status enable
set ssl-mirror enable
set ssl-mirror-intf "VirtualSnort" "VirtualPigPen" "VirtualSquid"
set ips-sensor "Just One Rule"
set ssl-ssh-profile "SSLIntercept"
```

Chris Farrell, chrisf@iquest.net
Monitoring systems detected and alerted on the test data just as the firewall had in Single Device Deployment testing:

```
chrisf@snot:/var/log/snot$ cat alert
[Priority: 0]
11/23-16:33:06.072450 172.16.1.23:58750 -> 208.88.250.45:443
TCP TTL: 63 TOS: ox0 ID: 27291 IpLen:20 DgmLen:390 DF
***AP*** Seq: 0x80441E1B Ack: 0x147B4228 Win: 0x13F TcpLen: 20

More importantly, network traces taken from the security monitoring systems provided confirmation of clear-text delivery to each system:
```

Chris Farrell, chrisf@iquest.net
# Upcoming Training

<table>
<thead>
<tr>
<th>Event</th>
<th>Location</th>
<th>Dates</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SANS London October 2020</td>
<td>London, United Kingdom</td>
<td>Oct 12, 2020 - Oct 17, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS October Singapore 2020 - Live Online</td>
<td>Singapore, Singapore</td>
<td>Oct 12, 2020 - Oct 24, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Dallas Fall 2020</td>
<td>Dallas, TX</td>
<td>Oct 19, 2020 - Oct 24, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS London November 2020</td>
<td>, United Kingdom</td>
<td>Nov 02, 2020 - Nov 07, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS San Diego Fall 2020</td>
<td>San Diego, CA</td>
<td>Nov 16, 2020 - Nov 21, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Frankfurt November 2020</td>
<td>, Germany</td>
<td>Nov 30, 2020 - Dec 05, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS Cyber Defense Initiative 2020</td>
<td>Washington, DC</td>
<td>Dec 14, 2020 - Dec 19, 2020</td>
<td>CyberCon</td>
</tr>
<tr>
<td>SANS OnDemand</td>
<td>Online</td>
<td>Anytime</td>
<td>Self Paced</td>
</tr>
<tr>
<td>SANS SelfStudy</td>
<td>Books &amp; MP3s Only</td>
<td>Anytime</td>
<td>Self Paced</td>
</tr>
</tbody>
</table>