jamk.fi

Outbound SSL/TLS decryption

Security impact of SSL/TLS interception

Joni Korhonen

Master's thesis
May 2019
School of Technology
Master's Degree Programme in Information Technology
Cyber Security

jamk.fi

Description

Author(s)	Type of publication	Date
Korhonen, Joni	Master's thesis	May 2019
		Language of publication: English
	Number of pages 62	Permission for web publication: x

Title of publication

Outbound SSL/TLS decryption

Security impact of SSL/TLS interception

Degree programme

Master's Degree Programme in Information Technology, Cyber Security

Supervisor(s)

Sampo Kotikoski

Assigned by

Government ICT Center (Valtori)

Abstract

There is more SSL/TLS encrypted traffic than ever. SSL/TLS encryption is designed to provide confidentiality, integrity and authentication; however with encrypted traffic there comes a challenge how to effectively monitor and control the transferring applications and data. Encrypted traffic could include malicious content or a risk company's business by e.g. leaking sensitive information. Decryption enhances security device's functionalities by providing more clear text content.

The objective was to research how outbound SSL/TLS decryption can be implemented, what to consider when implementing the decryption and to research how the decryption affects the cyber security domain in a lab environment.

The research was based on qualitative research method executed on a controlled lab environment with different case studies. The research was conducted with different research scenarios including malware and data loss in which outbound SSL decryption was used. Decrypted traffic findings were analyzed to classify traffic content and to find out if it was malicious.

The results describe the key concepts around SSL decryption and what to consider when planning or implementing outbound SSL decryption. The results included findings on decrypted payload, applications and decryption performance.

Organizations should take a closer look on decrypting outbound SSL/TLS traffic to remove a possibly existing security blind spot and to obtain more realistic situational awareness of cyber security. With SSL decryption, companies are balancing between privacy and security; however privacy is not equal to security.

Keywords/tags (SSL decryption, SSL interception, Cyber Security)

Miscellaneous

jamk.fi

Kuvaus

Tekijä(t)	Julkaisun laji	Päivämäärä
Korhonen, Joni	Opinnäytetyö, ylempi AMK	Toukokuu 2019
		Julkaisun kieli: Englanti
	Sivumäärä 62	Verkkojulkaisulupamyönnetty: x

Työn nimi

Ulospäin suuntautuvan SSL/TLS-liikenteen purkaminen

SSL/TLS-liikenteen purkamisen vaikutus tietoturvaan

Tutkinto-ohjelma

Master's Degree Programme in Information Technology, Cyber Security

Työn ohjaaja(t)

Sampo Kotikoski

Toimeksiantaja(t)

Valtion tieto- ja viestintätekniikankeskus (Valtori)

Tiivistelmä

SSL/TLS-salattua liikennettä on enemmän kuin koskaan. SSL/TLS-salaus on suunniteltu mahdollistamaan eheys, luotettavuus ja tunnistaminen, mutta salaus aiheuttaa myös haasteita. Liikenteen monitorointi, tietoturvakontrollien tehokkuus ja ympäristön realistisen tilannekuvan muodostaminen vaikeutuvat. Salatun liikenteen sisällä voi olla haitallista sisältöä tai riskejä yrityksen liiketoiminnalle. Salauksen purku tehostaa mahdollisesti olemassa olevien tietoturvalaitteiden toimintaa tarjoamalla enemmän salaamatonta sisältöä.

Tavoitteena oli tutkia, miten ulospäin suuntautuvan SSL/TLS-liikenteen purkaminen voidaan toteuttaa, mitä tulee ottaa huomioon purun käyttöönotossa ja miten liikenteen purkaminen vaikuttaa kyberturvallisuuteen testiympäristössä.

Tutkimus perustui laadulliseen tutkimusmenetelmään, joka toteutettiin kontrolloidussa testiympäristössä. Tutkimuksessa käytettiin useita kokeellisia testiskenaarioita, joissa testattiin SSL/TLS-purun konkreettisia vaikutuksia tietoturvakontrolleihin, monitorointiin, tilannekuvaan ja loppukäyttäjäkokemukseen. SSL/TLS-salattuja yhteyksiä käytettiin haittaohjelmien lataamiseen ja tiedon vuotamiseen ulos. Purettua liikennettä analysointiin, jotta saataisiin parempi käsitys, mitä sisältö oli ja oliko se haitallista.

Tutkimuksen tulokset auttavat ymmärtämään SSL/TLS-liikenteen purkamiseen liittyviä olennaisia teorioita ja käytännön asioita, jotka tulee ottaa huomioon purkamista käyttöönottaessa. Tuloksissa on analysoitu puretun liikenteen sisältöä ja purkamisen käyttöönoton yhteydessä tehtyjä huomioita.

Organisaatioiden tulisi tutustua tarkemmin SSL/TLS-liikenteen purkamiseen, jotta tietoturvan tilannekuvasta saadaan realistisempi ja tietoturvakontrollit tehostuvat.

Avainsanat (SSL decryption, SSL interception, Cyber Security)

Muut tiedot

Acronyms

(EC)DHE	(Elliptic-curve) Diffie–Hellman
3DES	Triple Data EncryptionAlgorithm
AD	Active Directory
AES	Advanced Encryption Standard
C	Client
C&C	Command and Control
CA	Certificate Authority
CBC	Cipher Block Chaining
CH	Client Hello
CN	Common Name
CR	Certificate Repository
CRL	Certificate Revocation List
DB	Database
DHE	Diffie-Hellman Ephemeral
DLP	Data Loss Prevention
DTLS	Datagram Transport Layer Security
ECDHE	Elliptic-curve Diffie–Hellman
EdDSA	Edwards-curve Digital Signature Algorithm
GCM	Galois/Counter Mode
HSTS	HTTP Strict Transport Security
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPS	Intrusion Prevention System
MD5	Message-Digest 5
NGFW	Next Generation Firewall
OCSP	Online Certificate Status Protocol
PKCS	Public Key Cryptography Standards
PKI	Public Key Infrastructure
PSK	Pre-shared key
RA	Registration Authority
RFC	Request for Comments
RSA	Rivest–Shamir–Adleman
RTT	Round-Trip-Time
S	Server
SH	Server Hello
SHA	Secure Hash Algorithm
SIEM	Security Information and Event Management
SOC	Security Operations Center
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TIA	TLS Intercept Application
TLS	Transport Layer Security

UDP	User Datagram Protocol
URL	Uniform Resource Identifier

Contents

Figures

1 Introduction

1.1 SSL/TLS traffic and decryption security aspect

SSL/TLS (commonly referred to as SSL) is an encryption protocol used in TCP/IP networks such as the Internet. It is a key element used by applications such as web-browsing, email, instant messaging and VoIP (Voice-over-IP). SSL/TLS combined with HTTP is called HTTPS (Hypertext Transfer Protocol Secure). HTTP and HTTPS are used between client web-browsers and web servers.

SSL traffic is now estimated to be over 72% of all network traffic. The amount has increased by almost 20% in one year. Usually encryption is preferred for security but it also presents big challenges to inspect traffic for threats. Cyber criminals use encrypted traffic to hide their presence and deliver or exfiltrate data. (Maddison 2018)

NSS Labs Inc. did a research in which they found HTTPS traffic increased over 90% in one year. More than 40% of websites encrypts traffic by default in July 2016 and it is predicted that in year 2019, 75% of all web traffic is encrypted (NSS Labs 2016). According to Google, 95% of traffic it encountered in August 2018 was encrypted. Of the unencrypted traffic, 86.4% was used by mobile devices. (Google 2018)

Encrypted traffic adds safety by ensuring confidentiality and integrity while at the same time hiding secured content from outsiders. However, on the other hand, encrypted traffic could include hidden threats so that detection systems such as Intrusion Prevension Systems (IPS) and Data Loss Prevention (DLP) cannot inspect encrypted content, which then can affect security monitoring and security controls.

According to Cyren’s security researchers, 37% of all malware found utilize HTTPS. They also stated that major ransomware families since January 2016 have used HTTPS for distibution. The malware use of HTTPS has increased dramatically by 30% in just six months in 2017. (Magnúsardóttir 2017)

Zscaler statistics stated that 60% of all detected threats used encryption in 2017. According to Zscaler, encrypted malicious content has more than doubled in six months in 2017. Cybercriminals use free SSL providers allowing them to bypass the

integrity checks and network monitoring tools of web browsers. Companies using less effective IP and domain block lists to identify malicious web sites clients could be compromised via legitimate website that includes malicious scripts in advertisements. Hackers also post phishing content from legitimate domains that they control. Zscaler detects approximately 10,000 SSL encrypted web exploits each month. Phishing attempts have increased by 400% from 2016 to 2017. Malware and botnets are increasingly using SSL traffic for callback home activities with an average of 5 000 Command and Control (C&C) attempts over SSL each day. (Desai 2017)

SSL can be used in different states and scenarios of attack: web-based compromise, C&C traffic, lateral movement and data exfiltration. Malicious SSL traffic includes e.g. exploit kits, malware, adware and malware callbacks. Typically, the outbound SSL traffic is decrypted using already existing Next Generation Firewall (NGFW) or web proxies. In the end, the decryption goal is not to spy on an employee but to implement better security with improved visibility and controls possibilities.

1.2 Research objective

The research objective of the thesis is to find out what security impacts intercepting and decrypting SSL traffic have and what needs to be concerned when implementing decryption of outbound SSL traffic. By decrypting SSL traffic it can be researched what is hidden in encrypted traffic payload that would otherwise bypass detection. Other decrypting devices such as Symantec SSL visibility were limited off the research scope since the objective of the research is to investigate the security impact of SSL decryption and not to compare the capabilities of different SSL decrypting devices. Setting up SSL decryption is an essential prerequirement for investigating SSL decrypted traffic, and it is included in the research with Palo Alto Networks NGFW. SSL traffic could include e.g. Trojans or Command and Control traffic.

1.3 Research method

Qualitative research method is used in this research. There is a Palo Alto Networks NGFW with outbound SSL decryption and Threat Prevention installed and configured in a controlled lab environment. The lab environment has live users and a Windows 10 workstation for different test scenarios. The research focuses on decrypted payload findings and sets up outbound SSL decryption with Palo Alto NGFW. The research needed to be done in lab environments for privacy and security policy reasons.

1.4 Research questions

1.4.1 Setting up SSL decryption

What needs to be noticed when intercepting outbound SSL traffic and setting up decryption? How much planning and administrator work is needed for setting up the system? Does SSL decryption affect user experience and web browsing delay?

1.4.2 Analyzing decrypted traffic

How is decrypted payload analyzed? What can be found in the lab environment with NGFW when doing SSL decryption?

1.4.3 Security impact

What is the security impact gained or lost from intercepting and decrypting SSL traffic? What possibilities does analyzing and controlling a decrypted payload provide? What would be the risks if SSL decryption had not been in place? Was the security blind spot removed by decrypting SSL traffic?

1.5 Thesis structure

This thesis consists of six parts including introduction, theory, research, research results, evaluation of results and conclusion. First, the introduction part explains the background of the subject and the scope of the research. The second part is about theory helping a reader with basic IT knowledge to understand the key concepts

about the subject. Chapter three explains how research is conducted and describes lab environment used for the research. The fourth chapter is about results that were found during the research. In the fifth chapter, the results of the previous chapter are evaluated. The last part discusses the conclusions of the thesis and answers the previously set research questions.

2 Theory on decrypting SSL/TLS

2.1 Background

Secure Sockets Layer (SSL) and its followers Transport Layer Security (TLS) versions 1.0, 1.1, 1.2 and 1.3 are cryptographic protocols to provide communications security and data integrity over TCP/IP networks such as the Internet. TLS encrypts end-to-end network connection in the transport layer. By design, TLS allows client server applications to communicate over networks without eavesdropping, message forgery or tampering. TLS provides communication confidentiality, data integrity and authentication over networks using cryptography. In typical web browsing use, TLS authentication is unilateral and only the server is authenticated, not the client. Hence, the client web browser validates the server's identity with a certificate; yet, from the server's perspective the client remains unauthenticated. (Rescorla 2018)

TLS uses both symmetric and asymmetric encryption. Asymmetric encryption have key pairs and it is used to authenticate the server and exchange the symmetric key. The symmetric key is used for data encryption and when data is encrypted with one key, it can only be decrypted by the other key in the pair. Advange of asymmetric encryption is that it allows to setup a secure transfer between peers without pre-existing secured communication. In TLS public asymmetric keys are used to eastablish symmetric keys. The disadvantage of using asymmetric keys is that it is slow for devices to perform. (PaloAlto Networks 2015a)

Symmetric key encryption can use a single key to encrypt and decrypt messages. Disadvantage is that the key must be known by both parties before any secure communication can start. This limitation makes it hard to use for quick needs and makes it impossible to securely send new key information if current key is compromised. (PaloAlto Networks 2015a)

2.2 TLS versions

TLS1.3 RFC8466 standard was introduced during the writing of this thesis. Following chapters introduce the reader to the new TLS version 1.3 and describe main

differences to previous TLS version. While everyone should use secured TLS versions 1.3 or 1.2, older versions are still used.

Figure 1 below shows a timeline of SSL/TLS versions. TLS has had problems with security over the last years, including Heartbleed, BERserk, Goto fail and more. Even when previous problems are not caused by the protocol design but lack of testing, TLS 1.3 design goals were to improve security and make the protocol faster. (Sullivan 2018)

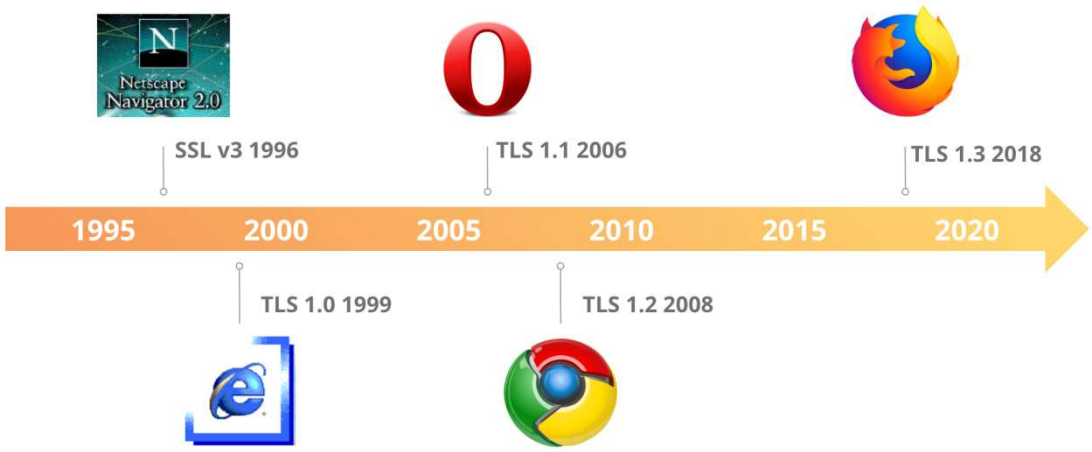


Figure 1. SSL/TLS versions releases timeline (Sullivan 2018).

While TLS1.3 is not backwards compatible with the previous versions, clients and servers can negotiate a version supported by both peers if the use of older TLS versions is allowed on both peers. Particularly higher layer protocols such as applications can use TLS transparently. (Rescorla 2018)

TLS 1.3 is a more secure protocol than the previous versions. It introduced encryption of handshake messages and removed older static cipher suites. TLS1.3 basic key exchange modes are PSK-only, PSK with (EC)DHE and (EC)DHE (Diffie-Hellman over either finite fields or elliptic curves). (Rescorla 2018)

TLS has a handshake protocol and a record protocol. The communicating parties are authenticated with handshake protocol and during the handshake cryptographic modes and parameters are decided. A record protocol uses the decided parameters to secure traffic between communicating peers. The record protocol flow consists of

fragmenting data into blocks, optional compression of data, adding message authentication code (MAC), encryption of data with negotiated cipher, perform padding if needed by cipher.

Datagram Transport Layer Security (DTLS) provides communication privacy for datagram protocols such as UDP while TLS is designed for TCP. DTLS is based on TLS protocol and is designed to prevent eavesdropping, message forgery and tampering. More information can be found in RFC6347. (Rescorla 2012)

2.3 TLS 1.3 key differences

In TLS 1.3 the whole handshake state machine is restructured to be consistent. After ServerHello message, all handshake messages are now encrypted and elliptic curve algorithms are now included in the base specifications. In addition to many minor differences, in TLS1.3 static RSA and Diffie-Hellman cipher suites has been removed. New signature algorithms such as Edwards-curve Digital Signature Algorithm (EdDSA) are now included. The redesign was carried out on key derivation functions, and zero RTT (Round Trip Time) was introduced to save a connection round trip for the application data. Other cryptographic improvements such as changing RSA padding to use RSA Probabilistic Signature Scheme are performed. (Rescorla 2018)

The design goals for TLS 1.3 were to reduce observable data, reduce session setup latency, address known (CBC, RC4) payload protection issues, reevaluate TLS handshake content and improve privacy e.g. padding and less long term-identifying values. (Roelof 2017)

In TLS1.3 different keys are used for each session so if man-in-the-middle breaks one session, the same key can not be utilized for decrypting other sessions.

Figure 2 illustrates a key difference of handshake performance when comparing TLS1.3 to older TLS versions. In TLS 1.2, a two-time round trip needs to be done before application data can be sent. In TLS 1.3 only one round trip is needed and 0-RTT data is also possible. (Jackson 2018)

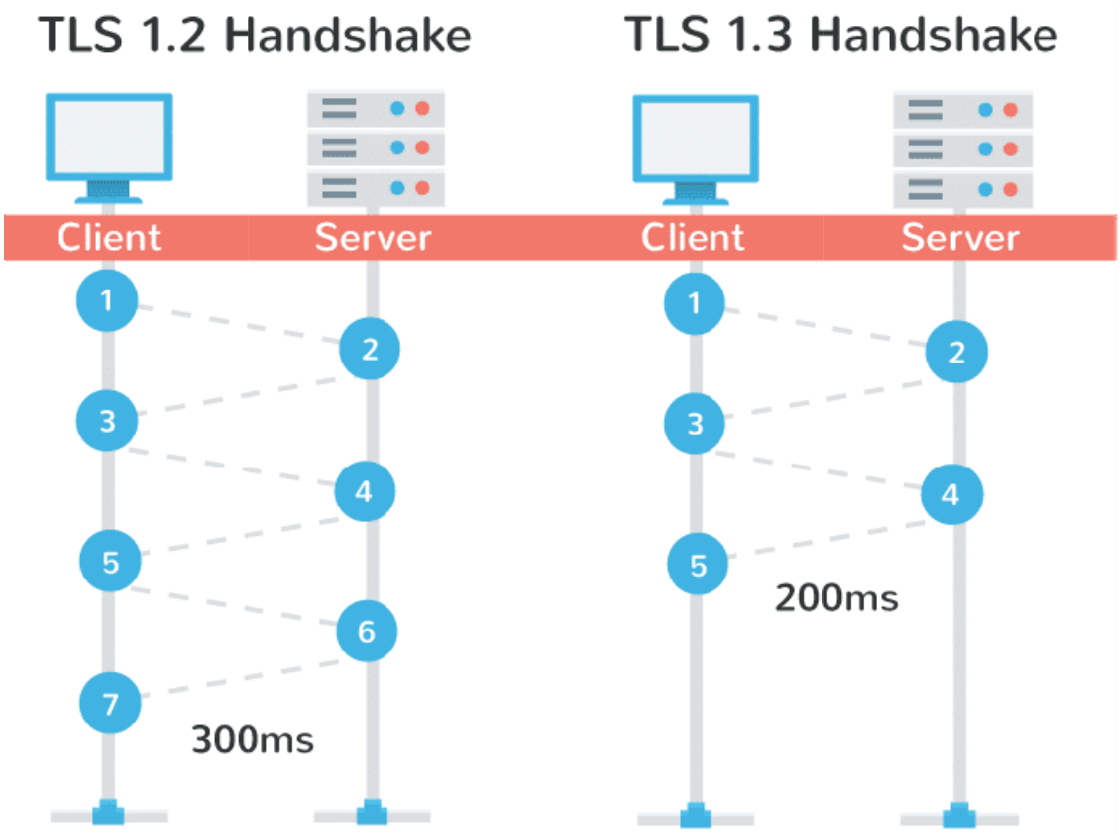


Figure 2.TLS 1.3 handshake performance (Jackson 2018).

With 0-RTT Data introduced in TLS 1.3, application level payload can be sent as part of the first flight of data from the client to the server. 0-RTT means that it does not require negotiation with the server before data can be sent. Early data traffic is encrypted with keys typically extracted from previous TLS1.3 session. (Roelof 2017)

2.4 Client certificate authentication

Client certificate authentication is a method to strengthen user authentication on a server. Authentication can be performed in several ways e.g. by asking for a user password, key, card, digital certificate or thumbprint. In this case the digital certificate is used to authenticate a user. A client digital certificate is usually a PKCS12 file loaded in to application. As seen in Figure 3, after the server has sent hello and its own certification, the server can optionally request client certification. This third optional step enables server to authenticate client. (Villanueva 2015)

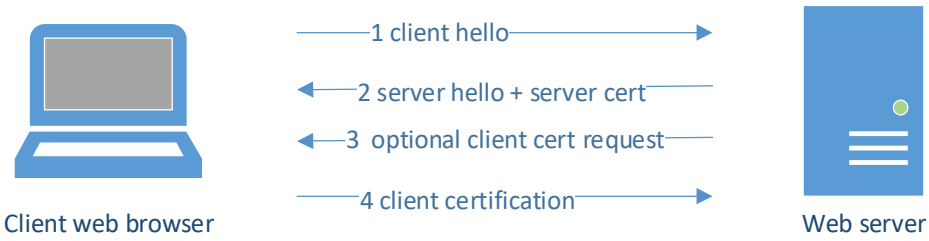


Figure 3. Client certificate authentication process

2.5 Public Key Infrastructure

SSL/TLS takes advantage of both symmetric and asymmetric encryption strengths. Public Key Infrastructure (PKI) is a method of packing and distributing asymmetric keys. Certificate is a key packaged with information about the key and signed with digital signature by the key authority. (PaloAlto Networks 2015a)

PKI is a framework that generates public and private key pairs and exchanges those. PKI is used for authentication, confidentiality, non-repudiation and integrity. Authentication means that the message is truly sent from the identity one is expecting. Confidentiality means that the message is encrypted. Integrity means that the message is not altered by unauthorized parties. (Cooper 2008)

The main components of PKI are Certification Authority (CA), Registration Authority (RA), Certificate Revocation List (CRL) and Certificate Repository (CR). CA issues certificates, signs private key certificates, binds user identity to public key and maintains certificate during its lifetime. RA accepts and verifies registration info,

accepts and authorizes requests but cannot issue certification. RA offloads requests for CA. CRL is a list of certificates that have been revoked and a method to tell not to trust that current public key. CRL is needed when e.g. users lose private key, people leave the company or their identity name changes. CRL is digitally signed by CA. CR contains the system that keeps all unexpired certificates and certificate information. Digital certificates currently use X.509 version 3. (Cooper 2008)

Online Certificate Status Protocol (OCSP) can be used by clients to check the revocation status of the authentication certificate. The client sends request with the serial number of the certificate to OCSP server. The server searches its database of the CA that issued the certificate and responds to the client the certificate status: good, revoked or unknown. OCSP can verify certificate status in real-time, instead of depending on the issue frequency of CRLs. (Palo Alto Networks 2019a)

Chain of trust is a list of certificates used to authenticate an entity such as a server. Chain of Trust includes Root CA, intermediate CA and SSL certificate. Root CA is self-signed certificate since the issuing authority is itself. Root CA is the basis of PKI deployments. (Palo Alto Networks 2015a)

PKI steps (Choi 2018) are listed below:

1) Sender asks the certificate directory for the receiver's public key

2) Publicly accessible directory that storages certificates send the key

3) Sender generates a session key, encrypts it with receiver public key, signs it with own private key and then sends it.

4) Receiver requests and validates the sender public key from the public directory.

5) Both parties trust each other and encrypt their message

2.6 Outbound SSL decryption

2.6.1 Intercepting SSL/TLS general

Outbound SSL decryption device acts as a man-in-the-middle box decrypting passing traffic and encrypting the client side with a different certificate. The device intercepts client SSL request, then forwards the request to the server but generates a certificate on-demand in response to the client's request. Resulting connection is between the client computer and the decrypting device. Decrypting device initiates another secure channel to the actual server using the server's certificate. Device acts as a forward proxy sitting middle of the two secure connections. (Palo Alto Networks 2015a)

The previous procedure is illustrated below in Figure 4.

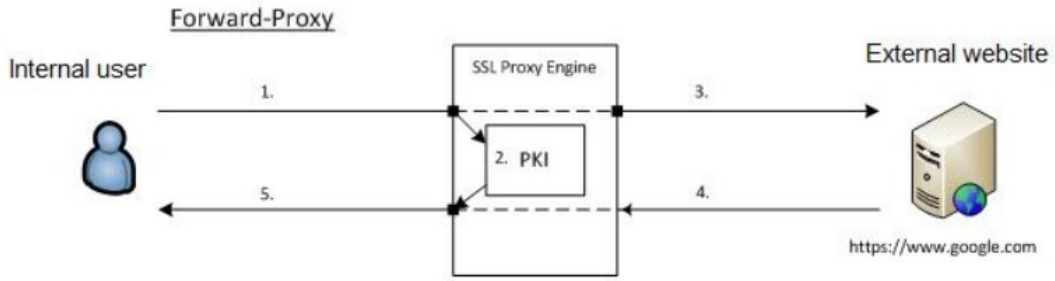


Figure 4. Outbound SSL decryption forward proxy (Palo Alto Networks 2019b)

The certificates used on client side are usually internal certificates that clients trust. By using trusted certificates the client does not receive notices of invalid certificates. The certificates generated for example by Microsoft AD can be installed to the operating systems of the client machine and imported into the decryption device. Some SSL traffic cannot be decrypted for several reasons: legal (privacy laws), there is not enough performance, or can not be implemented when client certificate authentication or pinned certificate is used. Depending on the decryption device it might handle client certificate authentication differently and allow configuration of the default behaviour when noticing client certification request.

By default, an intercepting device uses key size used by the destination server; however it can be configured to use a static key size. The device is only proxying SSL connections, not the underlying traffic. Some applications may not work with SSL forward proxy. These include apps that use client-side certificates, non-RFC compliant applications, server using unsupport crypthographics. (Palo Alto Networks 2015a)

2.6.2 Intercepting TLS 1.3

"Given that TLS intercept applications have a security purpose it should go without saying that those applications should not downgrade the security attributes of the TLS session. " (Roelof 2017)

Principles for TLS interception should include: (Roelof 2017)

- Do not downgrade the crypthographic strength
- Actively track, protect and fix against vulnerabilities
- Respect and follow regulations and privacy
- Validate certificate path
- Be secure by default

Figure 5 below illustrates a possible protocol flow when TLS Interception Appliance (TIA) intercepts Client (C) Server (S) TLS 1.3 session. (Roelof 2017)

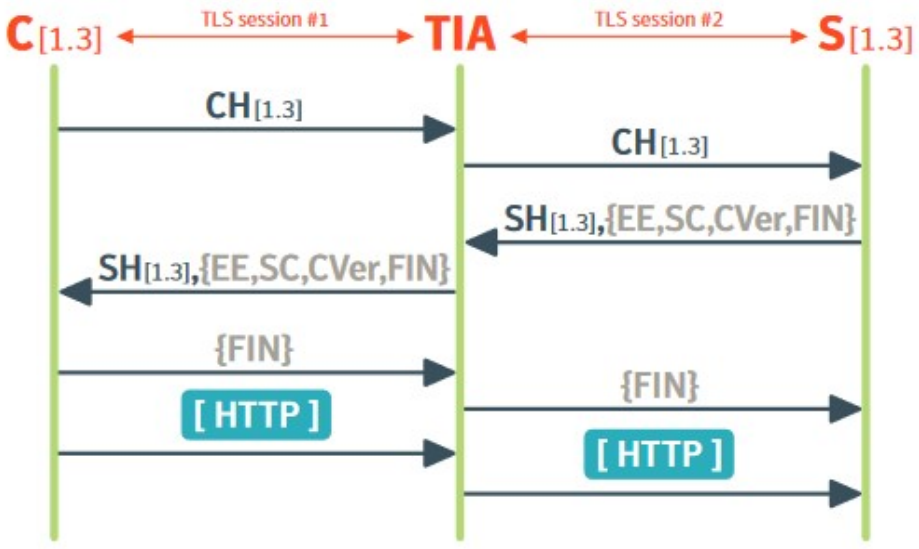


Figure 5. TLS 1.3 Intercept protocol flow (Roelof 2017)

In TLS1.3 application payload is encrypted with different keys than the handshake messages. TLS1.3 reduces handshake setup form 2-RTT to 1-RTT. TLS 1.3 does not need an explicit ChangeCipherSpec (CSS) signal to switch to the encrypted phase. An intercepted TLS 1.3 session would follow protocol flow illustrated below. TLS 1.3 adds a complication, ClientHello (CH) must include the (EC)DHE public value in the keyshare extension. TLS1.3 deprecates RSA key exchange and favors (EC)DHE, which means that TIA must be inline to participate in TLS handshake. (Roelof 2017)

Since TLS 1.3 introduced the 0-RTT data where early data is encrypted with traffic keys derived from PSK and the PSK is typically got from previous TLS 1.3 session. When TIA encounters TLS 1.3 early data on it could discard it; however to perform better TIA should forward early data to the server in the second session. Figure 6 below depicts a scenario where TIA can decrypt early HTTP data and then sends early data to server as part of the post-handshake application data. (Roelof 2017)

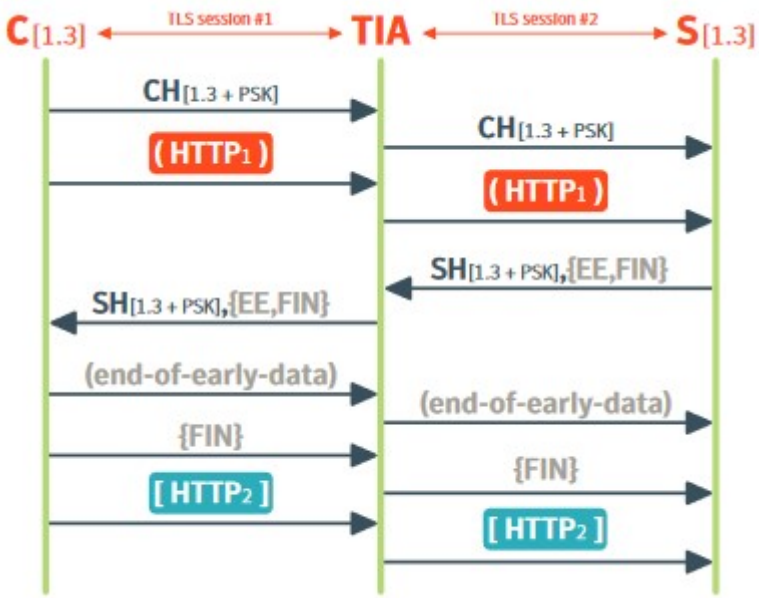


Figure 6. TLS 1.3 0-RTT Intercept with retain properties (Roelof 2017)

2.7 Certificate and public key pinning

HTTP Public Key Pinning (HPKP) associates a host with their expected certificate. Once the certificate for a host has been seen by the client, the certificate is associated or pinned to the host. There could be more than one certificate is allowed for the host. During the time of the pin user agents will require the host to present a certificate chain including at least one Subject Public Key Info structure whose fingerprint matches the pinned fingerprints user agent has on that particular host. By reducing the number of trusted authorities who can authenticate, pinning may reduce the man-in-the-middle attacks due to compromised Certification Authorities. (Evans 2015)

More information about public key pinning can be found in RFC7469.

2.8 HTTP Strict Transport Security

HTTP Strict Transport Security (HSTS) is an HTTP response header and when enabled HSTS informs web browsers that websites can only be accessed via HTTPS. Even when SSL certificates and HTTPS are setup correctly, HTTP is still available. HSTS forces the browser to load the secure version and ignores calls to open HTTP. HSTS also affects performance by removing a step in the process and allowing websites to load faster. More information on HSTS can be found in RFC6797. (Fuglseth 2018)

2.9 NGFW Content-ID

Palo Alto Networks Content-ID™ combines threat prevention engine with URL database and application identification to limit unauthorized data and file transfers. It detects and blocks malware, exploits and dangerous web surfing as well as targeted and unknown threats. (Palo Alto Networks 2016)

Content-ID is a stream-based not regular file-based architecture for real-time performance. Content-ID utilizes some of the same elements than Palo Alto Networks App-ID to limit unauthorized data, detect and block wide range of threats and control web surfing. Content-ID combines the threat prevention engine with URL database. Content-ID has ability to detect zero-day attacks with Wildfire sandboxing. The security profiles are: antivirus, anti-spyware, vulnerability protection, URL filtering, file blocking and data filtering. Antivirus detects infected files and anti-spyware detects spyware downloads and traffic from existing spyware. Vulnerability protection detects exploit attempts to known software vulnerabilities. URL filtering is used to control web browsing based on URL categories or URLs. File blocking tracks and blocks download and upload based on the file type and application. Data filtering is used as a DLP that looks for specific data patterns in traffic. (Palo Alto Networks 2016)

“Enterprise networks are facing a rapidly evolving threat landscape full of modern applications, exploits, malware and attack strategies that are capable of avoiding traditional methods of detection. Threats are delivered via applications that dynamically hop ports, use non-standard ports, tunnel within other applications routinely avoid proxies, and hide behind SSL or other types of encryption. These techniques can prevent traditional security solutions, such as IPS and firewalls from ever inspecting the traffic, thus enabling threats to easily and repeatedly flow across the network. Additionally, enterprises are exposed to targeted and customized malware, which may pass undetected through traditional antivirus solutions. Palo Alto Networks Content-ID addresses these challenges with unique threat prevention abilities not found in other security solutions. First, the next-generation firewall removes the methods that threats use to hide from security through the complete analysis of all traffic, on all ports, regardless of evasion, tunneling or circumvention

techniques. Simply put, no threat prevention solution will be effective if it does not have visibility into the traffic, and only Palo Alto Networks ensures that visibility through the identification and control of all traffic. Data filtering features enable administrators to implement policies that will reduce the risks associated with the transfer of unauthorized files and data." (Palo Alto Networks 2016)

Figure 7 below illustrates how Palo Alto Networks Content-ID workflow and different components (Palo Alto Networks 2016)

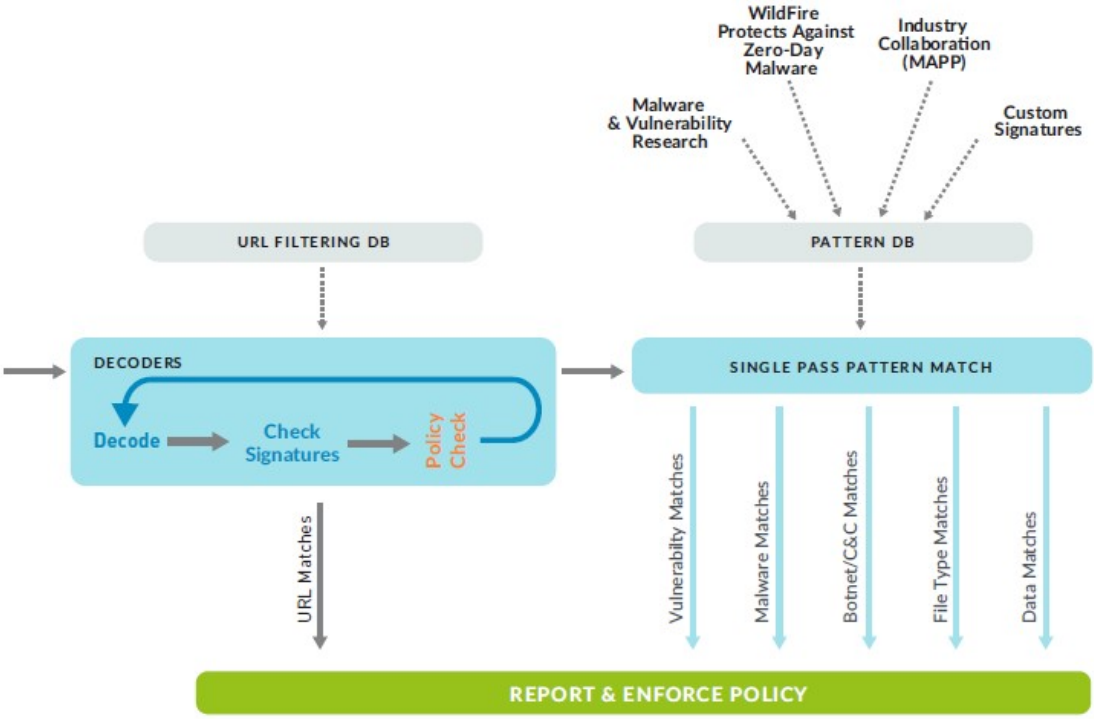


Figure 7. Palo Alto Networks Content-ID workflow (Palo Alto Networks 2016)

2.10 NGFW App-ID

Traditional firewalls are port-based firewalls that classify traffic by protocol and port. Today applications can easily bypass traditional port-based firewalls by hopping ports, using SSL or SSH. Palo Alto Networks App-ID is traffic classification mechanism that addresses the port-based firewall limitations. App-ID uses multiple mechanisms to determine correct identity of applications traversing the firewall. (Palo Alto Networks 2015a)

Applications are delivered through client-server model, web browser or decentralized peer-to-peer design. With Palo Alto solution an application is a specific program or feature that can be detected, monitored and blocked if needed. Applications can be business tools and needed services or personal services that need to be blocked in company environments. Applications have evasive capabilities meaning if e.g. a specific port is blocked it can try connecting with another port that could be open. (Palo Alto Networks 2015a)

If a zero day virus was using ports that are usually open, e.g. port 53 used for DNS, firewall would identify that it is not DNS traffic and it can be blocked and analyzed. When using UDP, a single packet contains the necessary info to identify the application. TCP protocol usually does not include all required information in any single packet for firewall to identify the application. In e.g. HTTP, after TCP handshake HTTP gets a request or a server reply includes application data for firewall to identify the application used. (Palo Alto Networks 2015a)

APP-ID uses multiple identification mechanisms to identify the application as demonstrated in Figure 8 below. Traffic is first classified based on the IP address and port. Signatures are applied to allowed traffic to identify the application based on application properties and transaction characteristics. If App-ID detects SSL or SSH is used and decryption policy is configured then traffic is decrypted and signatures are applied again on the decrypted flow. (Palo Alto Networks 2015a)

Protocol decoders are then used to apply context-based signatures to detect other applications that may be tunneling inside the protocol used, e.g. Facebook. For applications that are evasive and cannot be identified through advanced signature

and protocol analysis, heuristics and behavioral analysis may be used to determine the application identity. After the application has been identified, policy check determines how traffic is treated. (Palo Alto Networks 2015a)

Protocol decoders detect protocol in protocol within a session and provide a context for application signatures. Application signatures detect layer 7 signatures within a session and the used application signatures can be checked from the firewall application's page. Heuristics looks for pattern of communication when no signature exists. (Palo Alto Networks 2015a)

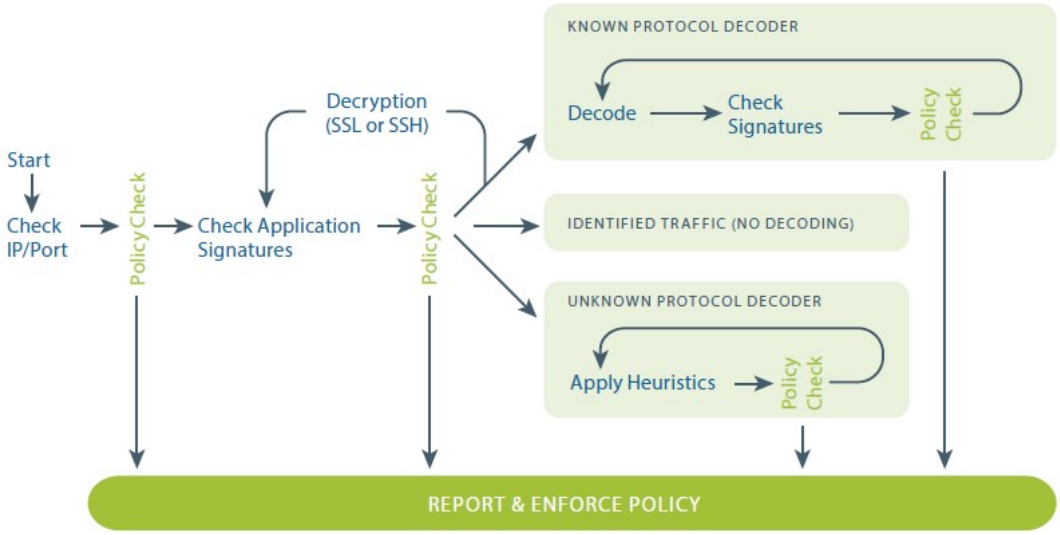


Figure 8. NGFW App-ID traffic identification workflow (Palo Alto Networks 2015b)

3 Research

3.1 Background and decryption implementation

In the research Palo Alto Networks NGFW was implemented for SSL decryption, performing application reqocnization and threat prevention with Palo Alto Networks Content-ID and App-ID techonologies, these functions are described earlier in theory chapter.

With Palo Alto Networks NGFW SSL decryption prerequisite is to have the firewall connected to network and to handle traffic. The initial installation and configuration of Palo Alto Networks firewall model PA-200 to the live lab environment is not described. In this environment PA-200 is used as a stateful layer 3 IP forwarding firewall to be able to decrypt outbound SSL traffic, analyze, capture and possibly act on the passing traffic. PA-200 firewall could also be installed in different modes; however, depending on the mode, the firewall might not be able to actively affect traffic flows.

The live lab environment and certificates used are presented in Figure 9. There is a Windows 10 PC with Mozilla Firefox and Microsoft Edge browsers. Palo Alto Networks PA-200 NGFW intercepts SSL requests so that it proxies the outbound SSL traffic and generates a certificate on the fly for the website the client is trying to connect with. After client requests SSL connection NGFW forwards request to the destination. When server sends it’s certificate that has been signed by public CA NGFW signs a copy of server certificate with it’s own CA. Different session keys are used between the client and the NGFW than the NGFW and the server.

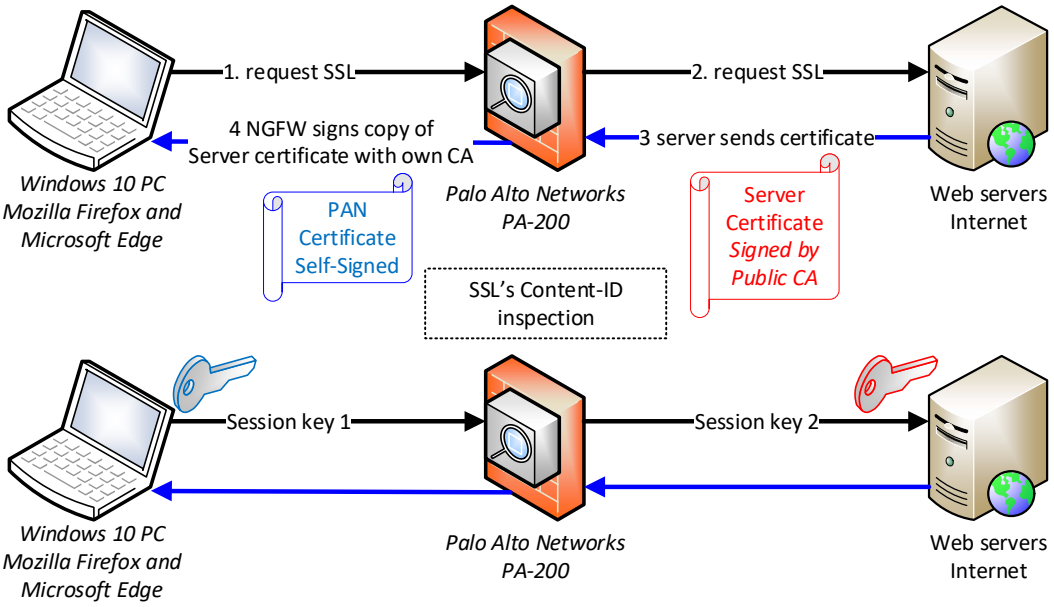
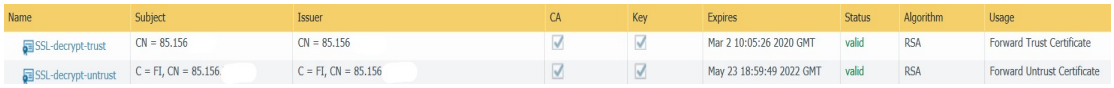


Figure 9. NGFW outbound SSL decryption and content inspection

After installation of the firewall, CA needs to be imported to the firewall or generated by the firewall. In the lab the firewall was used to generate the CA certificate. After that decryption profiles and decryption rules can be configured. In this lab, an optional SSL decryption notification page was used at start to inform end-users that SSL traffic is going to be decrypted. The firewall uses the current latest 8.1.6 software with scheduled updates that provide new threat intelligence from the vendor every minute.

NGFW generated certificate takes validity date for its certificate from the real server certificate. The issuing authority of the generated certificate in this case is the NGFW itself, which is seen in Figure 10. Subject and issuer fields are modified since those are the public IP address of the lab environment. The CA certificate generated by the firewall is used to encrypt SSL traffic between client PC and the firewall. Between the firewall and subject HTTPS Web Servers public certificate is used to encrypt SSL traffic. If a real server is using a certificate not trusted by the NGFW, then the certificate used for decryption can be a second certificate that is not trusted by the clients. With the use of a different certificate on untrusted configuration, the clients can still get a browser warning of a possible man-in-the-middle. CRL and OCSP,

described in theory part, were not configured in this testing environment to check certificate revocation.



Name	Subject	Issuer	CA	Key	Expires	Status	Algorithm	Usage
SSL-decrypt-trust	CN = 85.156	CN = 85.156	☑	☑	Mar 2 10:05:26 2020 GMT	valid	RSA	Forward Trust Certificate
SSL-decrypt-untrust	C = FI, CN = 85.156	C = FI, CN = 85.156	☑	☑	May 23 18:59:49 2022 GMT	valid	RSA	Forward Untrust Certificate

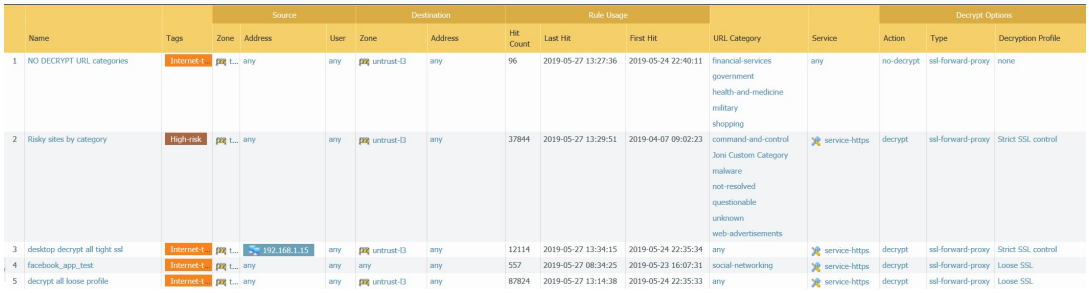
Figure 10. Firewall certificates

The web servers in Figure 9 refer to public HTTPS web servers in the lab's destinations. Public HTTPS web servers were narrowed by URL category, domains and exclusion list shown in Figure 11 and Appendix 1. Hence, only the traffic destined to a specific URL sites listed in custom URL category or vendor specified URL categories are decrypted if a domain is not listed in the vendor's predefined exclusion list.

The decryption policy in Figure 11 sets how decryption is implied. In decryption policies, five different rules were used. The first rule contains the action of no-decrypt on financial-services, health-and-medicine, government, military and shopping URL categories. These sites include e.g. Nordea and Amazon. Hence, this excludes the previously mentioned site categorizes and traffic destined to URL's belonging to those two specific categories are to bypass decryption. The second rule uses URL categories to match which website traffic to decrypt. Decryption policy strict SSL control is used for the risky sites and one workstation IP address. Fourth rule was done for other all other endpoint with social-networking URL-category for Facebook applications testing purposes. Last decryption rule included all other outbound HTTPS traffic to be decrypted with Loose SSL decryption profile.

Best practise is to start decryption with high priority URL categories that might include malicious content. In this lab configurations URL categories: command-and-control, malware, not-resolved, questionable, unknown, web-advertisements and author's own custom category are used. The content of author's custom category is shown in Appendix 2 and other URL categories get updated with Palo Alto Networks URL category updates. The current URL category content can be viewed from the firewall itself or from Palo Alto Networks website. If some website belongs to a

wrong category, the customer can make a request to Palo Alto Networks for URL category change.



			Source			Destination		Rule Usage					Decrypt Options		
	Name	Tags	Zone	Address	User	Zone	Address	Hit Count	Last Hit	First Hit	URL Category	Service	Action	Type	Decryption Profile
1	NO DECRYPT URL categories	Internet-t	t...	any	any	untrust-l3	any	96	2019-05-27 13:27:36	2019-05-24 22:40:11	financial-services government health-and-medicine military shopping	any	no-decrypt	ssl-forward-proxy	none
2	Risky sites by category	High-risk	t...	any	any	untrust-l3	any	37844	2019-05-27 13:29:51	2019-04-07 09:02:23	command-and-control Joni Custom Category malware not-resolved questionable unknown web-advertisements	service-https	decrypt	ssl-forward-proxy	Strict SSL control
3	desktop decrypt all tight ssl	Internet-t	t...	192.168.1.15	any	untrust-l3	any	12114	2019-05-27 13:34:15	2019-05-24 22:35:34	any	service-https	decrypt	ssl-forward-proxy	Strict SSL control
4	facebook_app_test	Internet-t	t...	any	any	any	any	557	2019-05-27 08:34:25	2019-05-23 16:07:31	social-networking	service-https	decrypt	ssl-forward-proxy	Loose SSL
5	decrypt all loose profile	Internet-t	t...	any	any	untrust-l3	any	87824	2019-05-27 13:14:38	2019-05-24 22:35:33	any	service-https	decrypt	ssl-forward-proxy	Loose SSL

Figure 11. Decryption policies

All decrypting rules are for HTTPS traffic towards internet zone untrust-l3; the decryption profiles used are more detailed in Figure 12, decryption type is ssl-forward-proxy, which means the firewall will proxy the outbound SSL traffic and action is set to decrypt. Figure 12 shows the decryption profiles that are used in the lab's previously mentioned decrypt rules and Appendix 11 more detailed view of the strict SSL control profile. In decryption profile one can configure decryption parameters that will be supported. Decryption profile settings define following parameters: supported key exchange algorithms, TLS versions, encryption algorithms and authentication algorithms.

After SSL traffic has been decrypted, it is then run through firewall policies that have the threat prevention capabilities antivirus, anti-spyware, vulnerability protection and data filtering enabled. These profiles inspect the decrypted SSL traffic payload and if for example a virus is seen, the firewall performs an action based on that profile specification. An optional notification page (in Appendix 3) was enabled so that the end user can receive a notification that SSL traffic will be decrypted. The notification helps the end user to understand that end to end privacy is not guaranteed.

Name	Location	SSL Forward Proxy			SSL Inbound Inspection		SSL Protocol Settings				No Decryption
		Server Certificate Verification	Unsupported Mode Checks	Failure Checks	Unsupported Mode Checks	Failure Checks	Key Exchange Algorithms	Protocol Versions	Encryption Algorithms	Authentication Algorithms	Server Certificate Verification
Loose SSL							RSA DHE ECDHE	Min Version: TLSv1.0 Max Version: Max	3DES RC4 AES128-CBC AES256-CBC AES128-GCM AES256-GCM	SHA1 SHA256 SHA384	
Strict SSL control		Expired Cert. Untrusted Issuers Unknown Cert Timeout Cert.	Version Cipher suite Client Auth.	No resources			RSA DHE ECDHE	Min Version: TLSv1.2 Max Version: Max	AES128-CBC AES256-CBC AES128-GCM AES256-GCM	SHA1 SHA256 SHA384	

Figure 12. Decryption profile

Figure 13 shows an example of browser information on secured https://www.iltalehti.fi website that is using TLS1.2 with AES 128 GCM encryption algorithm and SHA256 authentication algorithm. The key size used in encryption is 128bits. Browser trusted issuer for certificate is Amazon and the certificate will expire in 14.1.2020. Figure 17 on research results shows the same website's browser information when decrypted is enabled.

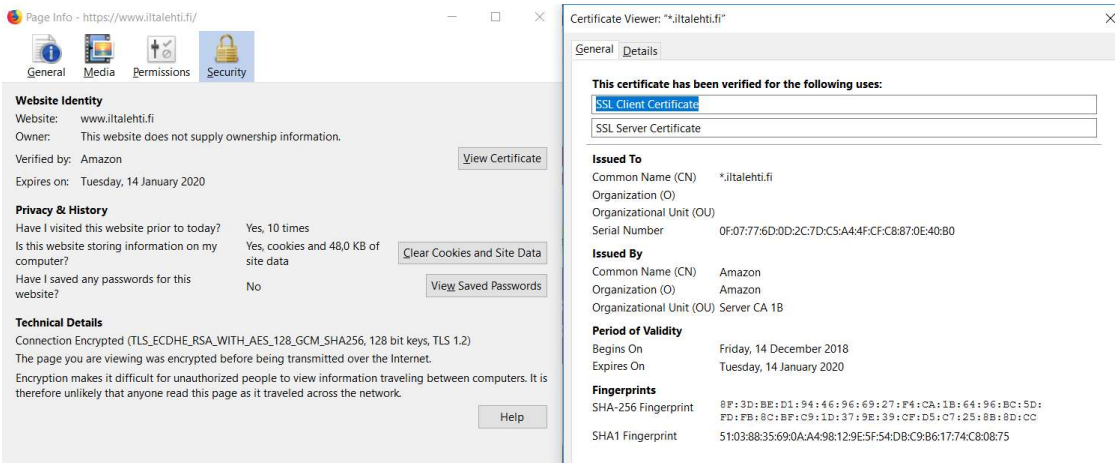


Figure 13. Encrypted website browser page info

3.2 End user experience and impact on TLS handshake

After SSL outbound decryption was enabled author and other lab users tested hundreds of decrypted websites in over 1 month test period. Author used strict SSL decryption profile testing while other users had loose SSL control in use. Loose SSL control allowed older TLS versions, more encryption algorithms and allowed other unsupported SSL to bypass encryption. Certificate handling was tested with www.badssl.com website.

Latency impact on SSL handshake and delay to HTTP transfer was also measured by using CURL command in Linux in the same network segment than the Windows 10

workstation. According to Palo Alto Networks knowledgebase article PA-200 should handle 1024 concurrent SSL decrypted sessions, see Appendix 10. For sizing or performance purposes reporting is used, in e.g. a report where all traffic destined to TCP port 443 from 7 days is included and the amount of decrypted traffic from 7 days. This is how the percentage of decrypted traffic can be calculated.

3.3 Administrative overhead

Work needed for setting up decryption was evaluated by author only. Author administrated decryption for 1 month test period to evaluate work needed to keep decryption working and ensure usability after implementation.

3.4 Security impact scenarios

3.4.1 Malware over SSL with threat prevention

Before starting a malware lab, the antivirus software had to be disabled from the author's computer because of the known well-known Eicar malware hashes and to make an exception with NGFW's URL filtering feature. URL filtering applied in security policies would have blocked the known Eicar malware website before SSL would have been used to transfer the malware to the computer. This also demonstrates the effectiveness of multi layered security that is also explained as onion layers of security.

In this lab test, Eicar malware test file download was tried over SSL encrypted session. Figure 14 illustrates the security policy configuration used for web browsing and threat prevention profiles that was selected. Threat prevention profiles antivirus, vulnerability protection, anti-spyware and file blocking allow NGFW to actively prevent threats entering e.g. internal clients behind the firewall.

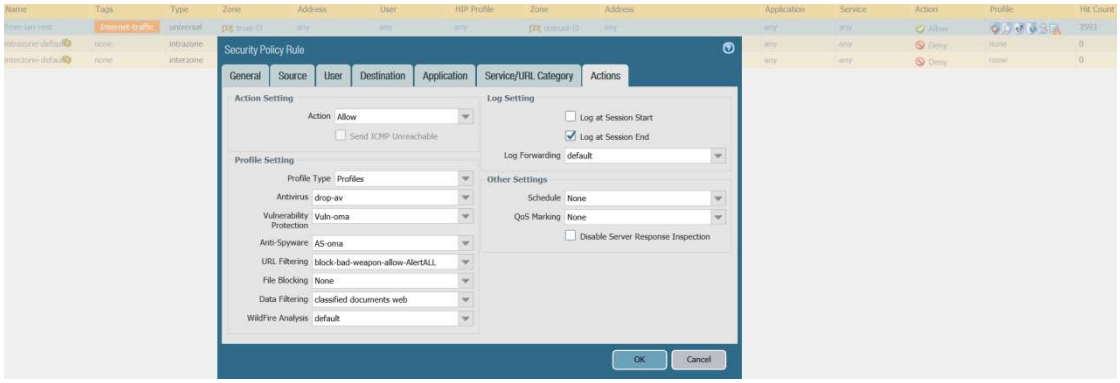


Figure 14. Security policy threat prevention profiles

3.4.2 Outbound SSL file transfer with DLP

In this lab “Classified” expression was used that was the indicator in the files to use the firewall’s data filtering. With DLP it is possible to prevent e.g. sensitive data from leaving the company. Data filtering is used in the lab to prevent classified test documents from leaking out of the environment via SSL connections. Figure 15 displays a data filtering profile that includes the specifications about applications and file types included in the analysis, direction of the traffic, alert and block thresholds, and log severity level if data leakage is seen. In this test file types selected were Microsoft Excel, Excel 97-2004, Powerpoint, Powerpoint 97-2004, Word, Word 97-2004 and Rich Text Format. Log severity was placed critical.

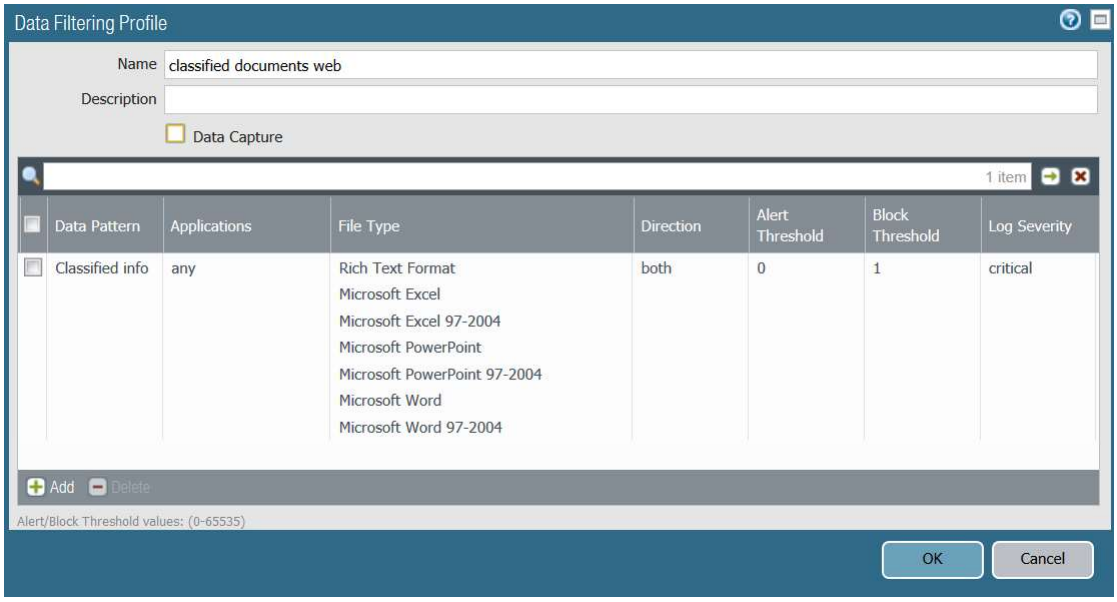


Figure 15. Data filtering profile

A data pattern configuration that is used by data filtering profile can be seen in Appendix 4. In the data pattern settings, “Classified” pattern was searched in multiple file types and fields. However a different pattern setting could have been used for each.

4 Research results

4.1 General results

During research it was noticed that Google Chrome and Internet Explorer use client machine trusted certificates; however, Mozilla uses its own certificate manager for trusted certificates. CA certificate generated by Palo Alto Networks Firewall as seen in Figure 10 was exported from the firewall and imported into client PC Mozilla Trusted Certificate Authorities, which is illustrated in Figure 16.

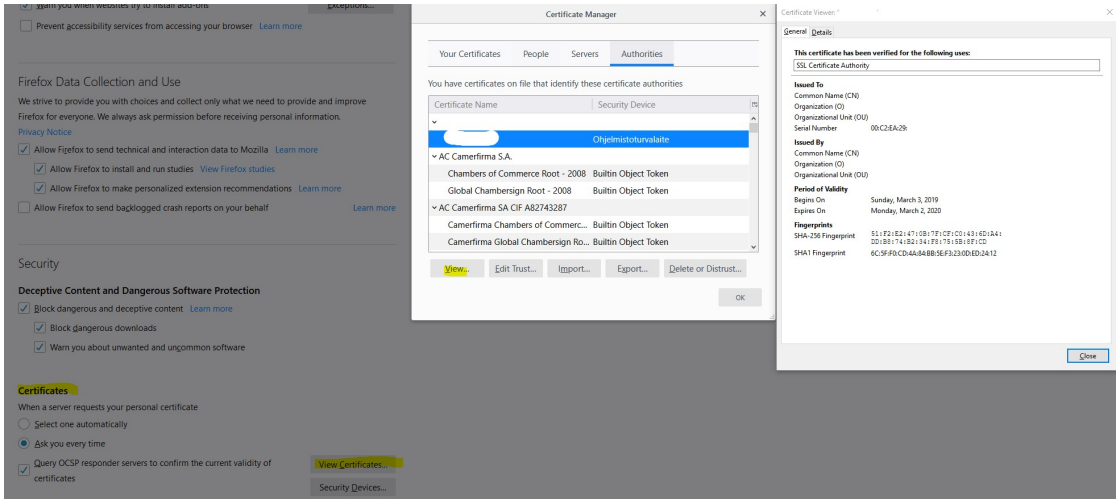


Figure 16. Mozilla Trusted Certificate Authorities

Earlier in Figure 13, there was an example with www.iltalehti.fi. Now with decryption enabled one can see that the issuer has been changed from Amazon to the author's own CN in Figure 17. Additionally, the client and server side still sees the same encryption parameters TLS 1.2 RSA with AES 128 GCM encryption algorithm and SHA256 authentication.

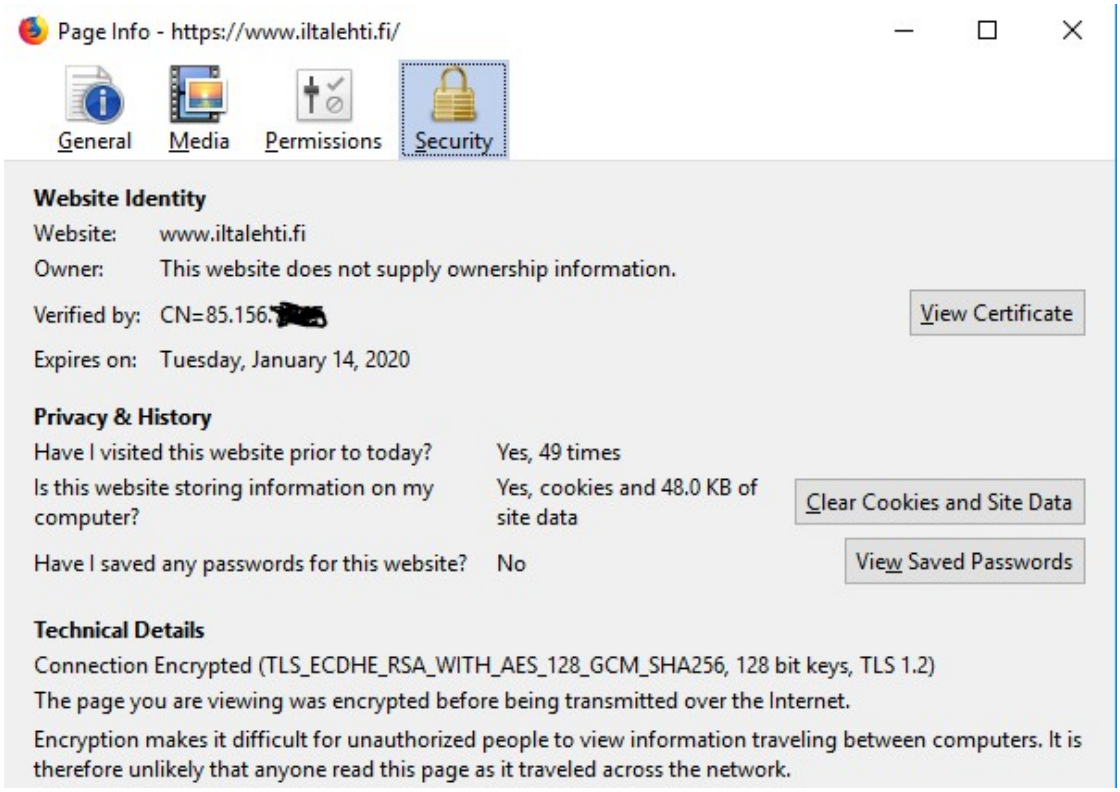


Figure 17. Decrypted website browser page info

There was no performance issues on 1 month test period with SSL decryption. All decrypted and non decrypted sites worked fine after installing firewall's generated certificate to all necessary clients or browsers. No end user usability issues or lowered performance were noticed by author or by any other lab environment user. There was no need for rebooting or restarting any processes during 1 month test period. Applications that fails to decrypt are cached and decryption is not tried again until 12 hours have passed.

The connections to decrypted HTTPS sites worked smoothly after implementation and decryption did not affect the round-trip-time noticeably. See Appendix 6 for the response measurements done with curl and Appendix 7 for firewall logs on these same tests. Appendix 12 shows www.badssl.com SSL certificate test site that was used to test certificate handling. Appendix 13-16 are listed the test results for certificate handling. The results were successful and according to the configuration used. Firewall's cached certificate and decrypted SSL session examples are shown in Appendix 8 and Appendix 9.

Appendix 17 illustrates a report example in which over 70% of sessions with destination port 443 was decrypted. Dataplane core was on low average load when decryption was enabled and traffic tests were ongoing, see Appendix 18. Some high load peaks were noticed but it did not affect user experience or performance. This could be tested further with traffic generators.

There was limitations with the firewall about decrypted data handling in port mirroring and packet captures. Packet capture functionality happens before decryption so no packet captures are available if threat prevention profiles does not trigger automatic packet captures such as single packet or extended packet captures. Use of decryption port mirroring that exports decrypted traffic flows to external devices is supported only on some virtual firewalls or bigger appliances than used in this lab environment.

Figure 18 displays a traffic log from the firewall that displays the decrypted traffic destined to search engines, Dropbox and other SSL sites. After decryption the real application is visible to the firewall's App-ID. This can be used for e.g. reporting or further security enhancements with application firewalling.

(flags has proxy) and (addr.src in 192.168.1.15)

Receive Time	Type	From Zone	To Zone	Source	Source User	Destination	To Port	Application	Decrypted	Action	Rule	Session End Reason	Bytes Sent	Bytes Received
04/07 10:54:21	end	trust-l3	untrust-l3	192.168.1.15		216.58.209.130	443	google-base	yes	allow	from-lan	tcp-fin	3.1k	6.7k
04/07 10:54:21	end	trust-l3	untrust-l3	192.168.1.15		216.58.209.130	443	google-base	yes	allow	from-lan	tcp-fin	3.1k	6.8k
04/07 10:51:11	end	trust-l3	untrust-l3	192.168.1.15		162.125.70.1	443	dropbox-editing	yes	allow	from-lan	tcp-fin	61.7k	82.0k
04/07 10:51:09	end	trust-l3	untrust-l3	192.168.1.15		162.125.70.1	443	dropbox-base	yes	allow	from-lan	tcp-fin	26.1k	36.0k
04/07 10:50:48	end	trust-l3	untrust-l3	192.168.1.15		162.125.70.6	443	dropbox-base	yes	allow	from-lan	tcp-fin	4.4k	13.1k
04/07 10:50:48	end	trust-l3	untrust-l3	192.168.1.15		162.125.70.6	443	dropbox-base	yes	allow	from-lan	tcp-fin	77.4k	12.4k
04/07 10:50:44	end	trust-l3	untrust-l3	192.168.1.15		162.125.70.1	443	dropbox-base	yes	allow	from-lan	tcp-fin	3.8k	7.3k
04/07 10:50:44	end	trust-l3	untrust-l3	192.168.1.15		162.125.70.1	443	dropbox-uploading	yes	allow	from-lan	tcp-fin	15.6k	21.6k
04/07 10:50:40	end	trust-l3	untrust-l3	192.168.1.15		13.107.42.12	443	ssl	yes	allow	from-lan	tcp-fin	541	9.0k
04/07 10:50:40	end	trust-l3	untrust-l3	192.168.1.15		13.107.42.12	443	ssl	yes	allow	from-lan	tcp-fin	601	9.0k
04/07 10:50:40	end	trust-l3	untrust-l3	192.168.1.15		13.107.42.12	443	ssl	yes	allow	from-lan	tcp-fin	541	9.0k
04/07 10:50:40	end	trust-l3	untrust-l3	192.168.1.15		13.107.42.12	443	ssl	yes	allow	from-lan	tcp-fin	541	9.0k
04/07 10:50:34	end	trust-l3	untrust-l3	192.168.1.15		13.107.42.12	443	ssl	yes	allow	from-lan	tcp-fin	541	9.0k
04/07 10:49:52	end	trust-l3	untrust-l3	192.168.1.15		172.217.21.162	443	google-base	yes	allow	from-lan	tcp-fin	5.0k	6.3k
04/07 10:48:29	end	trust-l3	untrust-l3	192.168.1.15		173.241.240.1...	443	web-browsing	yes	allow	from-lan-browsing-file-block	tcp-fin	3.7k	4.4k
04/07 10:48:29	end	trust-l3	untrust-l3	192.168.1.15		172.217.21.162	443	google-base	yes	allow	from-lan	tcp-fin	4.3k	5.8k
04/07 10:48:29	end	trust-l3	untrust-l3	192.168.1.15		172.217.21.162	443	google-base	yes	allow	from-lan	tcp-fin	2.3k	2.2k
04/07 10:48:29	end	trust-l3	untrust-l3	192.168.1.15		2.16.144.37	443	web-browsing	yes	allow	from-lan-browsing-file-block	tcp-fin	6.4k	9.7k
04/07 10:47:54	end	trust-l3	untrust-l3	192.168.1.15		35.157.96.7	443	web-browsing	yes	allow	from-lan-browsing-file-block	tcp-fin	4.5k	8.5k
04/07 10:47:09	end	trust-l3	untrust-l3	192.168.1.15		217.12.15.54	443	web-browsing	yes	allow	from-lan-browsing-file-	tcp-fin	5.1k	11.7k

Figure 18. Traffic log on decrypted traffic

4.1.1 Preventing malware over SSL

In test scenario, the client connects to SSL secured website and tries to download a malicious file. An optional notification page is configured and for that reason browser displays it and confirmation from user is needed to proceed. Figure 19 below

illustrates SSL inspection notification page and certificate information brief when decryption is enabled.

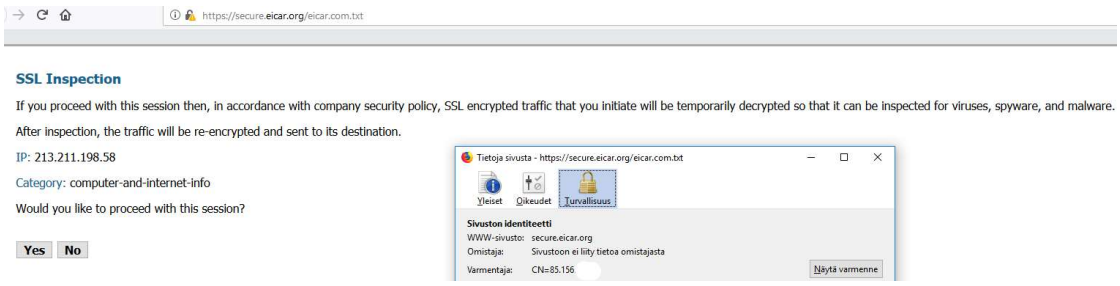


Figure 19. SSL Inspection notification page and certificate

When a user decides to continue forward and starts to download malicious content from Eicar website, the web browser notes that the connection to the page has failed. From firewall logs in Figure 20 can be seen that the threat log is now populated with Eicar File Detected and threat prevention did reset both server and client side of that TCP session.

Figure 20. NGFW file detected and reset client and server side

With this automatic preventing action client was protected even if client's antivirus software would not have noticed harmful file.

4.1.2 Classified information to public cloud data filtering

This scenario test was to leak classified information to public cloud and use firewall's data filtering or DLP function. During test it was noticed that Dropbox was using HSTS and when client did not have certificate trusted, Mozilla or IE did not let user to accept risk and proceed to Dropbox page, Figure 21 illustrates this.

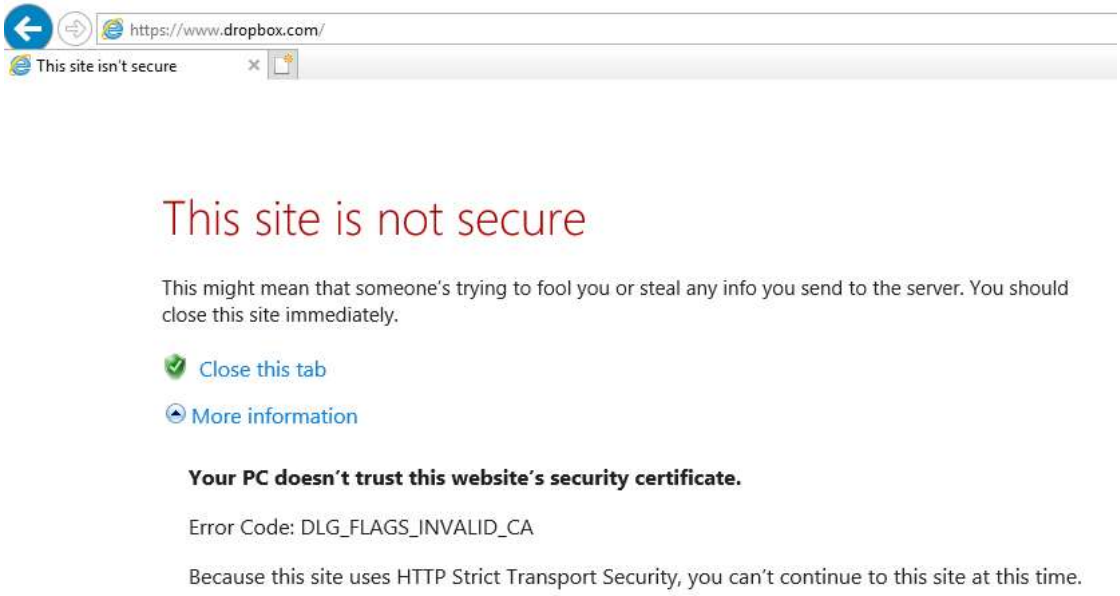


Figure 21. Dropbox uses HSTS

When self-signed certificate used in decryption was installed on Mozilla Trusted Certificate Authorities or Cert manager in Windows for Internet Explorer client could proceed to Dropbox. After trying to upload classified information to the public cloud data filtering prevented it with the earlier configured patterns as seen in firewall's traffic log Figure 22.

		Receive Time	Decrypted	Direction	Category	File Name	Name	From Zone	To Zone	Source address	Destination address	To Port	Application	Action
		04/07 11:08:03	yes	client-to-server	online-storage-and-backup	put_block_returning_token_unauth	Classified info	trust-l3	untrust-l3	192.168.1.15	162.125.70.6	443	dropbox-base	reset-server
		04/07 10:38:30	yes	client-to-server	Joni Custom Category	Classified information about innovation.doc	Microsoft Word DOC File	trust-l3	untrust-l3	192.168.1.15	104.25.154.6	443	web-browsing	alert
		04/07 10:38:30	yes	client-to-server	Joni Custom Category	Classified information about innovation.doc	Microsoft MSOFFICE	trust-l3	untrust-l3	192.168.1.15	104.25.154.6	443	web-browsing	alert
		04/07 10:38:30	yes	client-to-server	Joni Custom Category	Classified information about innovation.doc	Microsoft Word DOC File	trust-l3	untrust-l3	192.168.1.15	104.25.154.6	443	web-browsing	alert

Figure 22. Data filtering logs

4.2 Research discoveries

Configuring and implementing outbound SSL decryption with Palo Alto Networks Firewall did take only a couple hours and administrative overhead in 1 month test period was minor. Only fixes that was needed to make were installing firewall's certificate to all lab environment clients. Decryption policies did not need any more configurations during tests.

Best practise is to use category based decryption where categories such as unknown and known-bad websites are decrypted. The vendors categorize different websites to

different categories, e.g. Financial, Health and medicine. One could also exlude individual domains from decryption. It is possible to configure firewall to allow traffic without decryption or block sites that are using client certificate authentication or sites that have legal restriction for decryption. Appendix 16 illustrates that after importing Badssl client certificate to Mozilla and using strict SSL control, client certification authentication fails because of configurations in SSL profile. But when using Loose SSL profile, connection works – both client and server verified signed with DigiCert CA. Networks might also include self-signed or expired certificates. Decryption policies can define what to do with different certificates status, to e.g. block connections or to allow with untrusted certificate warning.

Palo Alto NGFW listens on all ports and is able to decrypt SSH, SSL inbound and SSL outbound traffic. Palo Alto Networks NGFW supports RSA, DHE, (EC)DHE key exchange algorithms and 3DES, RC4, AES128-CBC, AES256-CBC, AES128-GCM, AES256-GCM encryption algorithms. Authentication algorithms supported are MD5, SHA1, SHA256, SHA384. NGFW supports following SSL protocol versions: SSLv3.0,TLS v1.0-v1.2.

If SSL traffic cannot be decrypted or is using outdated risky algorithms, it is possible to drop that specific traffic to gain higher security. Chrome and some other browsers establish sessions using QUIC which has proprietary encryptions that firewall can't decrypt instead of using SSL/TLS. QUIC can be blocked in configurations. When trying to decrypt sites that block decryption it results in blocking that traffic.

When traffic is SSL/TLS encrypted application identification gets more complicated but decryption enables identification of actual application running over SSL/TLS. Figure 23 demonstrates that in the firewall application SSL can now be seen as different Facebook applications enabling companies to block in e. g. chatting, file-sharing but still allowing work related applications Facebook-base and social media posting.

Source	Source User	URL Category	Destination	To Port	Application	Decrypted	Action	Rule	Session End Reason	Bytes Sent	Bytes Received
192.168.1.15		social-networking	157.240.194.35	443	facebook-posting	yes	allow	from-lan	tcp-fin	6.9k	15.1k
192.168.1.15		social-networking	31.13.72.8	443	facebook-chat	yes	allow	from-lan	tcp-fin	22.1k	18.1k
192.168.1.15		social-networking	31.13.72.12	443	facebook-base	yes	allow	from-lan	tcp-fin	6.6k	244.4k

Figure 23 Traffic log Facebook applications

5 Evaluation of results

5.1 SSL decryption capabilities

NGFW at the current 8.1.6 software negotiates server to use TLS1.2 instead of TLS1.3 and then decrypts TLS1.2 traffic. If TLS1.3 is the only option on the HTTPS server when using Palo Alto Networks decryption the NGFW cannot decrypt the traffic with the current 8.1.6 software. Some decrypting devices e.g. Symantec SSL visibility currently supports TLS1.3, see Appendix 5.

During the 1 month test period no problems were found caused by decryption. There were no performance issues and regular usability was confirmed from the test environment users. Decryption was successfully implemented even when using the vendor's smallest appliance PA-200 and decryption was started quite heavily on multiple used URL categories.

5.2 Setting up decryption system

NGFW SSL decryption initial setup was fast and easy in the lab environment. In bigger production environments the decryption needs to be started carefully and small scale. Decrypting should start with the risky URL categories or URLs and after that monitor a cool down period if end users report errors. Additionally policies what to do with self-signed or expired certificates should be defined. With Palo Alto Networks solution and category based decryption by design customer trusts vendor's categories.

Depending on the environment, it can take some work to deliver and install the trusted certificates to all client devices if there already is no internal CA in use. If there is an existing CA certificate that clients trust, one can make a certificate request and import a CA certificate to the decrypting device. If client does not trust a certificate and the web sites are using HSTS, the clients will not be able to continue browsing even when trying to accept a warning. The use of HPKP seems to be deprecated since Chrome also removed HPKP from version 68.

Different SSL profiles should be thought carefully concerning what kind of profile and settings to use with different URL categories or individual domains. A more strict profile should be used with potentially harmful or unknown sites while business needs and trusted sites could be allowed with a more loose SSL profile if needed.

Use of forward untrust certificate is recommended as it enables clients to get warnings about possible man-in-the-middle.

5.3 SSL decryption administration

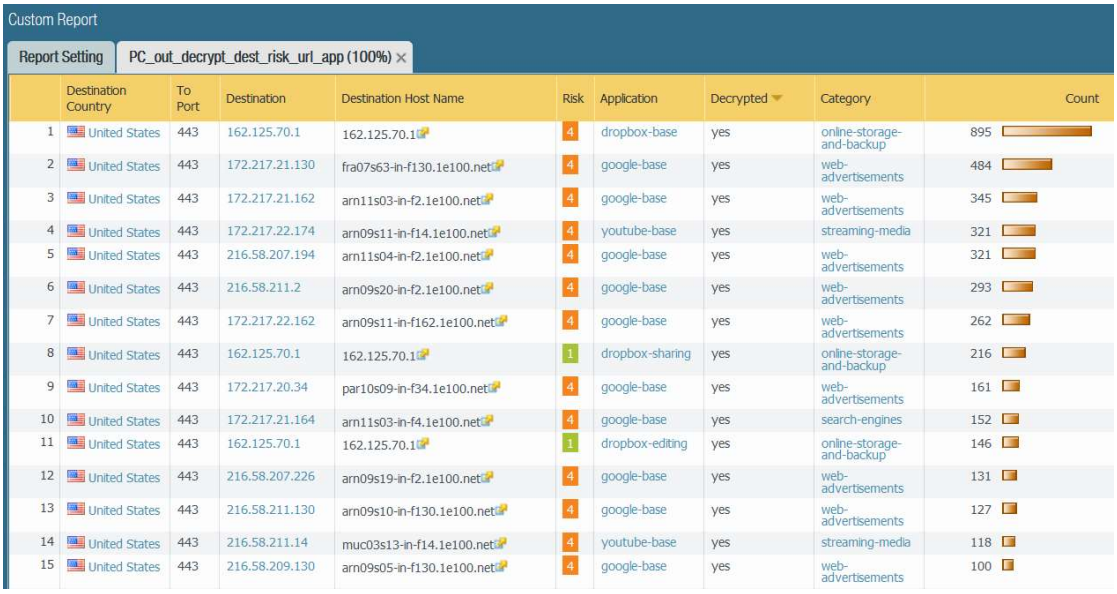
Administering SSL decryption with Palo Alto NGFW was straight forward and it did not increase administration work significantly. The use of vendor provided URL categories is essential so that the requirements for both privacy regulations and effective decryption administration can be met. Provided categorization intelligence reduces administration overhead by dramatically reducing the need to manually define static URLs to be decrypted or not to be decrypted. Without categorization intelligence the administrators would have to specify statically websites that were to be decrypted or try to decrypt all and add exceptions to websites that cannot be decrypted or blocked. Use of dynamic categories still enables flexibility to add those static entries or exceptions if needed.

When decrypting it should be noted that a user's private data will become visible. From the privacy perspective, it was a benefit that the firewall model used in the lab could not view or export out the decrypted payload. Only an export was possible if automated prevention module captured a packet. Otherwise, the same privacy concerns apply than before decryption. With Palo Alto log masking, it is possible to hide user identifying data so that daily firewall administrators do not see where and what each individual user is communicating. If SSL payload is visible as clear text, privacy concerns should be dealt with in a proper method before implementation.

5.4 Reports, logging and monitoring on threats

Figure 24 shows a report generated from the NGFW a month after the decryption was implemented. The report included logs from the decrypted traffic with the highest count of sessions. The vendor's risk categorizes these mainly with category 4

that is the second highest risk category. The report shows the actual applications instead of SSL to get more precise and realistic overview.



Custom Report

Report Setting | PC_out_decrypt_dest_risk_url_app (100%)

	Destination Country	To Port	Destination	Destination Host Name	Risk	Application	Decrypted	Category	Count
1	United States	443	162.125.70.1	162.125.70.1	4	dropbox-base	yes	online-storage-and-backup	895
2	United States	443	172.217.21.130	fra07s63-in-f130.1e100.net	4	google-base	yes	web-advertisements	484
3	United States	443	172.217.21.162	arn11s03-in-f2.1e100.net	4	google-base	yes	web-advertisements	345
4	United States	443	172.217.22.174	arn09s11-in-f14.1e100.net	4	youtube-base	yes	streaming-media	321
5	United States	443	216.58.207.194	arn11s04-in-f2.1e100.net	4	google-base	yes	web-advertisements	321
6	United States	443	216.58.211.2	arn09s20-in-f2.1e100.net	4	google-base	yes	web-advertisements	293
7	United States	443	172.217.22.162	arn09s11-in-f162.1e100.net	4	google-base	yes	web-advertisements	262
8	United States	443	162.125.70.1	162.125.70.1	1	dropbox-sharing	yes	online-storage-and-backup	216
9	United States	443	172.217.20.34	par10s09-in-f34.1e100.net	4	google-base	yes	web-advertisements	161
10	United States	443	172.217.21.164	arn11s03-in-f4.1e100.net	4	google-base	yes	search-engines	152
11	United States	443	162.125.70.1	162.125.70.1	1	dropbox-editing	yes	online-storage-and-backup	146
12	United States	443	216.58.207.226	arn09s19-in-f2.1e100.net	4	google-base	yes	web-advertisements	131
13	United States	443	216.58.211.130	arn09s10-in-f130.1e100.net	4	google-base	yes	web-advertisements	127
14	United States	443	216.58.211.14	muc03s13-in-f14.1e100.net	4	youtube-base	yes	streaming-media	118
15	United States	443	216.58.209.130	arn09s05-in-f130.1e100.net	4	google-base	yes	web-advertisements	100

Figure 24. NGFW report with top 15 decrypted applications

Palo Alto could add a clear view on what certificates firewall has seen proxying SSL traffic since it would be useful for an overall view of the certificates used in the environment. Some competitor's product has a view displaying what certificates are seen in the data path. It gives complete information what certificates the device has seen in the network, e.g. how much self-signed or expired certificates are seen.

5.5 Decrypted payload

Nothing alarming was found after extensively enabling outbound SSL decryption in the live lab environment. For testing purposes malware samples were accessed via HTTPS on Eicar website. It was discovered that the malware downloaded via HTTPS was previously not blocked by the firewall and after enabling SSL decryption, the firewall could see the decrypted payload and block the malware from entering the client machine.

The second case in a live lab environment used DLP with decryption. Dropbox was listed in URL category for decryption. Without the SSL decryption, classified information could have leaked to public cloud storages or e.g. via Facebook file

sharing to a competing company. But with decryption leaking data was filtered. Companies could add a security classification in document templates and specify DLP to find that value from the correct field, this way preventing valuable information from leaking out accidently or with rogue purposes.

5.6 Summary of results

There is manageable amount of work to design and decide before SSL decryption is running in a live production environment. Decryption on network layer also involves endpoint configurations with certificates and knowledge on different browser and additional security functions such as client certificate authentication. Using automatic threat prevention methods and actions reduces the administrators' need to monitor and response to observations. When planning on implementing decryption sizing should be carefully estimated.

Without SSL decryption the network security devices would have seen the size of the payload, used bandwidth, source and destination IP address, port and protocol. With SSL decryption enabled the firewall was able to stop the loss of sensitive data, identify and stop threats in content, stop transfer of specific file types e.g. EXE, identify data type sent, identify application and identify if security policy was violated.

The lab environment had benign SSL traffic since there could not be seen any malicious contents after enabling SSL decryption except the contents in test scenarios. Threats seen after enabling SSL decryption were the malware test files from Eicar and the lab's classified documents trying to leak outside.

TLS 1.3 makes intercepting encrypted traffic more difficult than before, not all SSL intercepting and decrypting devices yet support TLS 1.3. Both TLS endpoint application vendors and vendors making TLS interceptors must follow RFC standard to ensure interoperability.

6 Conclusions and discussion

6.1 General

Why decrypt traffic? SSL sessions might be hiding risky application signatures or hide sensitive data in file transfers. Decryption is used to gain more visibility, control and granular security. With SSL decryption, companies are balancing between privacy and security. The privacy concern needs to be taken care of before starting to decrypt traffic. How, where, who and why decrypted data is handled needs to be discussed. However, privacy does not equal security.

In the network topology designs, the decryption device placement and role need to be thought carefully so that the device gets all intended traffic and is located at the best possible point of the network. The live lab environment used in the research proved to be secure even after decryption was implemented, at least from the perspective of security devices used. The decryption did not have a negative effect on user experience and implementing SSL decryption is recommended by author.

SSL decryption vendors and resellers can have commercial agenda for providing statistics about SSL used for malicious intensions; however, a common trend can be seen in the statistics that points out the need for decryption. If SSL decryption had not been in place for the lab environment with the preventing security controls, malicious content and rogue intensions would have not been noticed on network level and malware been delivered to the endpoint.

There is plenty of work to plan, decide and configure before SSL decryption is in effective use. After implementation SSL decryption adds some administrator overhead depending on the environment. If the environment is quite static, meaning a company's managed clients use a limited amount of well-known secured public internet services, SSL decryption does not introduce significant administrator overhead. If client machines are managed by the company, the distribution of the trusted certificate is easier and does not require administration knowledge from end users. New cryptographic protocol TLS 1.3 brings challenges to SSL/TLS intercept vendors and TLS endpoint application developers to responsible handle TLS interception, so that privacy and security is not compromised. Interception appliance

vendors should and endpoint application developers should be involved with TLS standard designers. Only outbound SSL decryption was included in the research. Companies should also consider decrypting inbound SSL and decrypt SSH tunnels.

Companies cannot simply block all outbound SSL traffic and there might be harmful content encrypted. The blocking based on URLs and URL categories is a good idea but usually not manageable or difficult to implement. Unknown sites are not always possible to block by default and it brings administrator overhead to always manually add new sites to allowed list. SSL decryption brings possibilities to look inside encrypted traffic to identify what is the actual application and what data is transferred. Since the amount of encrypted traffic is growing, detection systems, forensic and analytic tools can lose a big part of their value without decryption. At least on one level comprehensive security controls and clear text content should align. In critical or high security environments there should be more than one layer of controls also on decrypted traffic.

In the endpoints, servers and in some cases load balancers' SSL connections are terminated and security solutions can be implemented without the need for separate decryption. Devices that do not have proper endpoint protection enabled need to be handled in the network layer. In addition to modern endpoint and server protection, comprehensive network security with SSL traffic handled secure is needed to provide better situational awareness of cyber security and comprehensive security.

6.2 Summary and future thoughts

Implementing network security today is more difficult than ever since the amount of encrypted traffic has increased lately in most industries and globally. With today's networks, encrypted traffic becomes a challenge for network security devices. In order to utilize the devices more effectively, SSL decryption should be considered. SSL decryption removes significant blind spot in today's network security and enhances the existing network security devices by decrypting payload, providing view of the actual application and payload. Not only active security controls such as IPS can get more value out of decryption but also forensic, analytic and reporting tools get a more detailed and realistic view.

NGFWs are typically used for consolidating security functions and to preserve SSL's promise of confidentiality. It can be more managable to meet compliance requlations, have less administration overhead and if necessary, with some appliances broker decrypted traffic to other devices e.g. analytic or forensic tools. In some cases, a separate decryption device might provide better cost/performance ratio since the hardware is optimized for decryption and all the resources are dedicated to decryption and not shared with other high value features. However, then it could be an additional device on the datapath that could break up and cause disruption.

Automated prevention products with fine-tuned configuration for the environment, trained administrators, strong partners and solid security architecture scoped for the business needs can affect the cyber security domain heavily. Nowadays trending SIEM and SOC products and services have their need; however, for most companies the previously mentioned key factors should be the ones to focus first. There are also solutions that include behavior or heuristic based analyzes and machine learning algorithms that do not require SSL decryption to define certain SSL traffic malicious. To make sure traffic content is benign, decryption and technical security controls aligned is a must.

Depending on the technical environment and business, SSL decryption can have major impacts on security. Statistics from several security vendors state that encrypted traffic is a real security concern. Outbound SSL decryption also supports building more realistic situational awareness of cyber security. IT service providers and enterprises should at least take a closer look at outbound SSL decryption and think how it could affect their security blind spot.

The author will continue working on TLS 1.3 decryption, certificate revocation subjects, using environment's root CA for decrypt certificate signing and adding sensitive destinations of traffic to bypass decryption if needed.

References

Choi, K. 2018. Introduction to PKI (Public Key Infrastructure). Accessed 4.5.2019. Retrieved from https://medium.com/@kennch/introduction-to-pki-public-key-infrastructure-e7863c9232f9

Cooper, D. 2008. IETF RFC5280. The Transport Layer Security (TLS) Protocol Version 1.3. Accessed on 24.3.2019. Retrieved from https://www.ietf.org/rfc/rfc5280.txt

Desai, D. 2017. Zscaler SSL/TLS-based malware attacks. Accessed 1.2.2019. Retrieved from https://www.zscaler.com/blogs/research/ssltls-based-malware-attacks

Evans C. 2015. Public Key Pinning Extension for HTTP. Accessed on 24.3.2019. Retrieved from https://tools.ietf.org/html/rfc7469

Fuglseth B. 2018. What Is HSTS and How Does It Protect HTTPS From Hackers? Accessed on 24.3.2019. Retrieved from https://www.makeuseof.com/tag/what-is-hsts/

Google, 2018. Transparency report: HTTPS Encryption on the web. Accessed 1.2.2019. Retrieved from https://transparencyreport.google.com/https/overview?hl=en

Jackson, B. 2018. An Overview of TLS 1.3 – Faster and More Secure. Accessed 1.4.2019. Retrieved from https://kinsta.com/blog/tls-1-3/

Maddison, J. 2018. Fortinet. More Encrypted Traffic Than Ever. Accessed 24.9.2018. Retrieved from https://www.fortinet.com/blog/industry-trends/more-encrypted-traffic-than-ever.html

Magnúsardóttir, A. 2017. Cyren Security Blog. Accessed 10.9.2018. Retrieved from https://www.cyren.com/blog/articles/over-one-third-of-malware-uses-https

NSS Labs. 2016. NSS Labs Predicts 75% of Web Traffic Will Be Encrypted by 2019. Accessed 15.3.2018. Retrieved from https://www.nsslabs.com/company/news/press-releases/nss-labs-predicts-75-of-web-traffic-will-be-encrypted-by-2019/

Palo Alto Networks. 2015a. PAN-EDU-201 Course Material

Palo Alto Networks. 2015b. APP-ID Tech bried. Accessed 24.3.2019. Retrieved from https://www.paloaltonetworks.com/resources/techbriefs/app-id-tech-brief

Palo Alto Networks. 2016. Content-ID tech brief. Accessed on 24.3.2019. Retrieved from https://www.paloaltonetworks.com/apps/pan/public/downloadResource?pagePath=/content/pan/en_US/resources/techbriefs/content-id-tech-brief

Palo Alto Networks. 2019a. Online Certificate Status Protocol (OCSP). Accessed on 24.3.2019. Retrieved from https://docs.paloaltonetworks.com/pan-os/8-0/pan-os-

admin/certificate-management/certificate-revocation/online-certificate-status-protocol-ocsp.html#

Palo Alto Networks. 2019b. Knowledgebase article. Accessed on 24.3.2019. Retrieved from https://knowledgebase.paloaltonetworks.com/KCSArticleDetail?id=kA10g000000ClV8CAK

Palo Alto Networks. 2019c. Knowledgebase article. Accessed on 24.3.2019. Retrieved from https://knowledgebase.paloaltonetworks.com/KCSArticleDetail?id=kA10g000000ClEZCA0

Rescorla E. 2012. RFC6347. Datagram Transport Layer Security Version 1.2. Accessed on 24.3.2019. Retrieved from https://tools.ietf.org/html/rfc6347

Rescorla, E. 2018. IETF RFC8446. The Transport Layer Security (TLS) Protocol Version 1.3. Accessed on 24.3.2019. Retrieved from https://tools.ietf.org/html/rfc8446

Roelof, D. 2017. Symantec. Responsibly Intercepting TLS and the Impact of TLS 1.3. Accessed 4.5.2019. Retrieved from https://www.symantec.com/content/dam/symantec/docs/other-resources/responsibly-intercepting-tls-and-the-impact-of-tls-1.3-en.pdf

Sullivan, N. 2018. A Detailed Look at RFC 8446 (a.k.a. TLS 1.3). Accessed 04.05.2019. Retrieved from https://blog.cloudflare.com/rfc-8446-aka-tls-1-3/

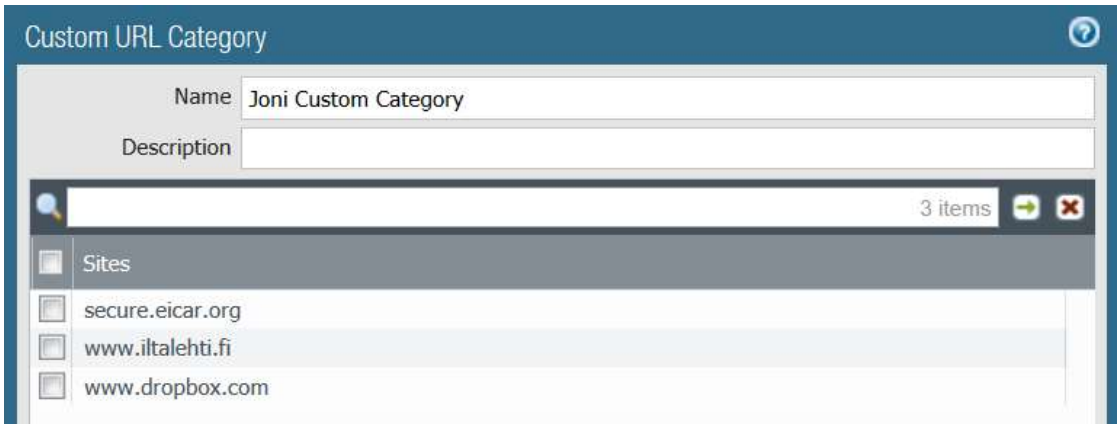
Symantec. 2018. SSLV 4.3.1.1 Release Notes. Accessed 12.11.2018. Retrieved from https://support.symantec.com/en_US.html

Villanueva, J. 2015. What is Client Certificate Authentication? Accessed on 24.3.2019. Retrieved from https://www.jscape.com/blog/client-certificate-authentication

Appendices

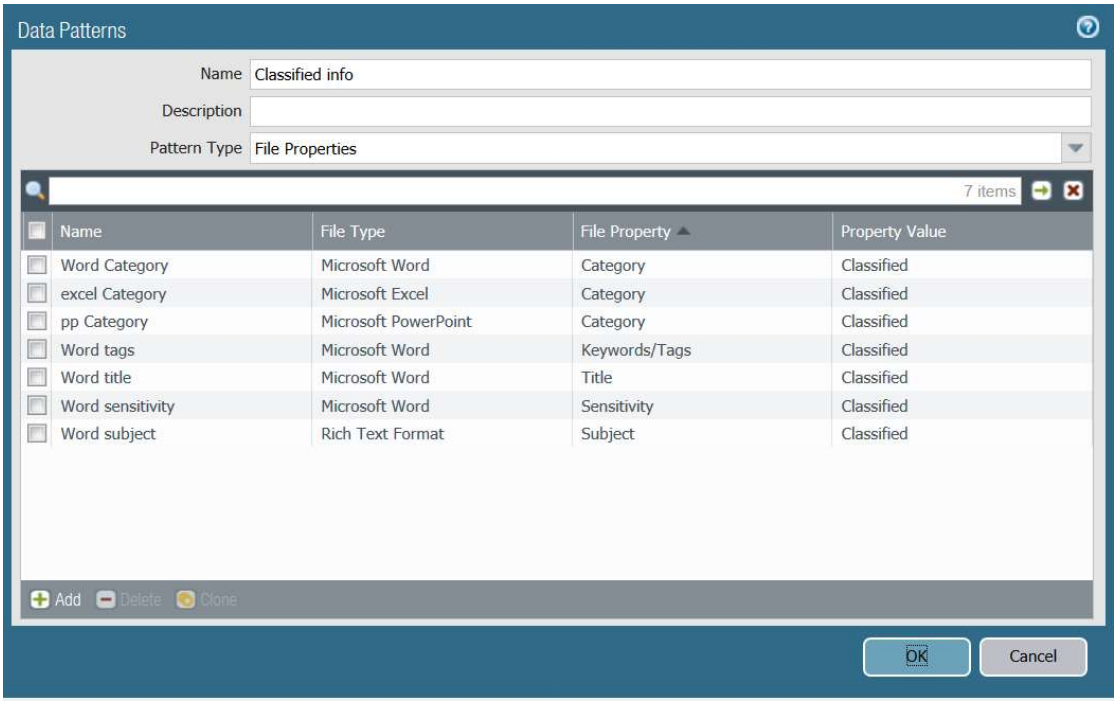
Appendix 1. SSL decryption exclusions

Appendix 2. NGFW author's custom URL category



Appendix 3. NGFW SSL decryption notification page

Appendix 4. NGFW data pattern configuration



Appendix 5. Symantec SSL visibility decryption capabilities

“SSL Visibility 4.2.5.1 includes the following new feature:
Support added for TLS 1.3 drafts 25-28
SSL Visibility 4.2.4.1 includes the following new features:
Path MTU Discovery support for ProxySG enabled and Classic segments
Support for TLS 1.3 drafts 18-24
SSL Visibility 4.2.1.1 provides the following new features.
TLS 1.3 Support - SSL Visibility 4.2.1.1 or later supports decryption and inspection of TLS 1.3 (drafts 18 - 21)
sessions on Classic segments. On ProxySG mode segments, TLS 1.3 traffic is automatically downgraded to TLS
1.2 for SSL Offload. Policy rules can be configured to cut through TLS 1.3 traffic without downgrading for interception
by the ProxySG device if desired. The following TLS 1.3 cipher suites are supported.
-TLS_AES_128_GCM_SHA256
-TLS_AES_256_GCM_SHA384
-TLS_CHACHA20_POLY1305_SHA256
-TLS_AES_128_CCM_SHA256
-TLS_AES_128_CCM_8_SHA256

CCM Cipher-Suite Support - SSL Visibility 4.2.1.1 adds support for the following cipher-suites.
-TLS_RSA_WITH_AES_128_CCM
-TLS_RSA_WITH_AES_256_CCM
-TLS_DHE_RSA_WITH_AES_128_CCM
-TLS_DHE_RSA_WITH_AES_256_CCM
-TLS_RSA_WITH_AES_128_CCM_8
-TLS_RSA_WITH_AES_256_CCM_8
-TLS_DHE_RSA_WITH_AES_128_CCM_8
-TLS_DHE_RSA_WITH_AES_256_CCM_8
-TLS_ECDHE_ECDSA_WITH_AES_128_CCM
-TLS_ECDHE_ECDSA_WITH_AES_256_CCM
-TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8
-TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8

Learned Certificate Cache - The SSL Visibility appliance has a new segment option to cache visiblecertificates used in SSL connections in the Learned Certificate Cache. Certificates are associated withdestinations, as defined by a Server Name Indicator (SNI), IP address, and port number, and areretrieved for policy evaluation when a certificate is encrypted or otherwise not available (as is the case forTLS 1.3 flows which always have encrypted certificates). The Learned Certificate Cache option isenabled by default. When a segment is initially created, the cache is empty but the SSL Visibilityappliance builds the cache as it sees X.509 server certificates to a particular destination.By maintaining a cache of the unencrypted SSL server certificate for destinations, SSL Visibility is able touse this information when making policy decisions even when the flow is a TLS 1.3 flow.” (Symantec 2018)

Appendix 6. SSL handshake and delay to transfer after decryption

curl -k -w "dns_resolution: %{time_namelookup}, tcp_established: %{time_connect}, ssl_handshake_done: %{time_appconnect}, TTFB: %{time_starttransfer}\n" -o /dev/null -s **"https://secure.eicar.org"**

dns_resolution: 0.029, tcp_established: 0.075, **ssl_handshake_done: 0.172, TTFB: 0.245**

curl -k -w "dns_resolution: %{time_namelookup}, tcp_established: %{time_connect}, ssl_handshake_done: %{time_appconnect}, TTFB: %{time_starttransfer}\n" -o /dev/null -s **"https://secure.eicar.org"**

dns_resolution: 0.029, tcp_established: 0.074, **ssl_handshake_done: 0.158, TTFB: 0.237**

curl -k -w "dns_resolution: %{time_namelookup}, tcp_established: %{time_connect}, ssl_handshake_done: %{time_appconnect}, TTFB: %{time_starttransfer}\n" -o /dev/null -s **"https://www.iltalehti.fi"**

dns_resolution: 0.029, tcp_established: 0.042, **ssl_handshake_done: 0.094, TTFB: 0.228**

curl -k -w "dns_resolution: %{time_namelookup}, tcp_established: %{time_connect}, ssl_handshake_done: %{time_appconnect}, TTFB: %{time_starttransfer}\n" -o /dev/null -s **"https://www.iltalehti.fi"**

dns_resolution: 0.029, tcp_established: 0.043, **ssl_handshake_done: 0.096, TTFB: 0.214**

Appendix 7. SSL handshake and transfer delay tests firewall log

(addr.src in 192.168.1.13) and (flags has proxy)

Receive Time	Type	From Zone	To Zone	Source	Source User	URL Category	Destination	To Port	Application	Decrypted	Action	Rule	Session End Reason	Bytes Sent	Bytes Received
05/21 21:52:42	end	trust-l3	untrust-l3	192.168.1.13		Joni Custom Category	13.32.56.64	443	web-browsing	yes	allow	from-lan-browsing-file-block	aged-out	12.1k	484.8k
05/21 21:52:24	end	trust-l3	untrust-l3	192.168.1.13		Joni Custom Category	13.32.56.20	443	web-browsing	yes	allow	from-lan-browsing-file-block	tcp-fin	12.4k	483.9k
05/21 21:51:33	end	trust-l3	untrust-l3	192.168.1.13		Joni Custom Category	213.211.198.58	443	web-browsing	yes	allow	from-lan-browsing-file-block	tcp-fin	1.6k	4.2k
05/21 21:51:30	end	trust-l3	untrust-l3	192.168.1.13		Joni Custom Category	213.211.198.58	443	web-browsing	yes	allow	from-lan-browsing-file-block	tcp-fin	1.6k	4.2k

Appendix 8. Display firewall certificate cache

show system setting ssl-decrypt certificate-cache

global trusted, hits: 20, refs: 1,

Root CA: 0079_DigiCert_Global_Root_CA.cer

original cert len 2275

subject *.adnxs.com

CRL OCSP status: valid, timeout(secs): 0

original serial number(16)

0a 04 09 13 2a 89 8c 8f 66 02 02 0a 59 be 6d 4d ....*... f...Y.mM

built x509 certificate

version 2

cert algorithm 4

valid 1901230000000Z -- 2103081200000Z

cert pki 2

subject: *.adnxs.com

issuer: 85.156.79.67

serial number(16)

7d 1e df 5a 29 c7 20 93 66 02 02 0a 59 be 6d 4d }..Z). . f...Y.mM

ec key size 256 bits siglen 256 bytes

basic constraints extension CA 0

global trusted, hits: 1, refs: 1,

Root CA: 0079_DigiCert_Global_Root_CA.cer

original cert len 2923

subject *.rubiconproject.com

CRL OCSP status: valid, timeout(secs): 0

original serial number(16)

03 17 b8 3e f4 9c d9 06 9b 14 37 ca 69 f3 58 40 ...>.... ..7.i.X@

built x509 certificate

version 2

cert algorithm 4

valid 1901100000000Z -- 2101141200000Z

cert pki 1

subject: *.rubiconproject.com

issuer: 85.156.79.67

serial number(16)

69 7e df 72 c7 da ed 73 9b 14 37 ca 69 f3 58 40 i~.r...s ..7.i.X@

rsa key size 2048 bits siglen 256 bytes

basic constraints extension CA 0

Appendix 9. Example of firewall cached sessions

show system setting ssl-decrypt session-cache

hosts (client/server) id/ticket age cipher_c cipher_s

192.168.1.15-->147.75.83.1 2360e236... 361 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

192.168.1.15-->173.241.240.143 55f9ea1f... 362 TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

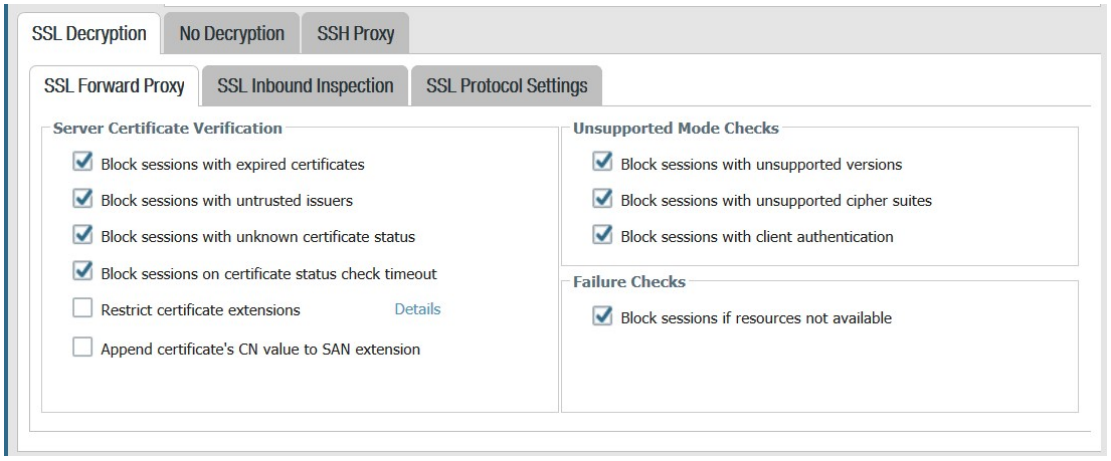
192.168.1.15-->69.173.144.165 2f95218c... 362 TLS_RSA_WITH_AES_256_CBC_SHA TLS_RSA_WITH_AES_256_CBC_SHA

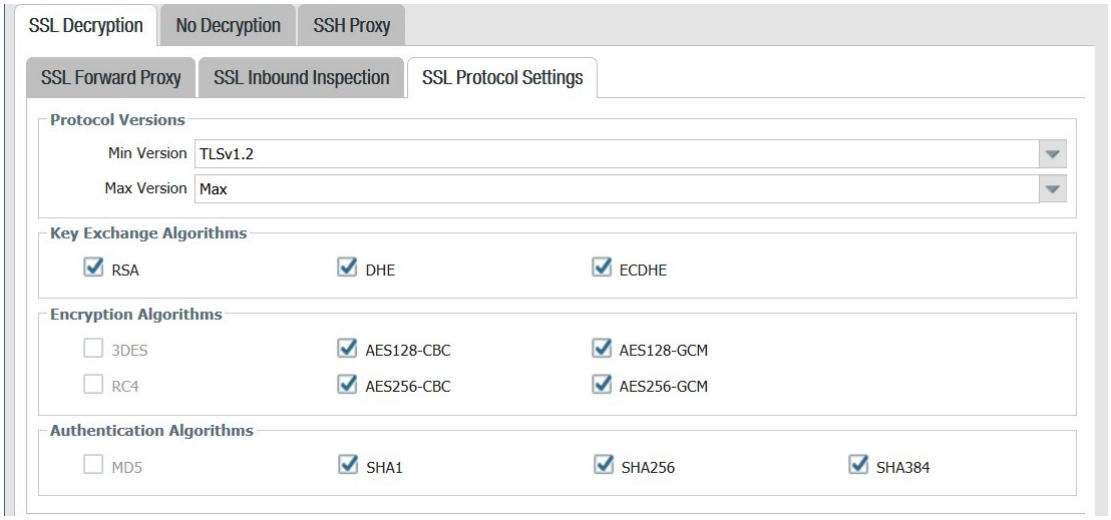
Appendix 10. SSL decryption session limitations per firewall model

Hardware	SSL Decypted Session Limit
VM-100	1,024 sessions
VM-200	1,024 sessions
VM-300	1,024 sessions
PA-200	1,024 sessions
PA-500	1,024 sessions
PA-2020	1,024 sessions
PA-2050	1,024 sessions
PA-3020	7,936 sessions
PA-3050	15,360 sessions
PA-3060	15,360 sessions
PA-4020	7,936 sessions
PA-4050	23,808 sessions
PA-4060	23,808 sessions
PA-5020	15,872 sessions
PA-5050	47,616 sessions
PA-5060	90,112 sessions
PA-7000-20G-NPC	131,072 sessions
PA-7050	786,432 sessions

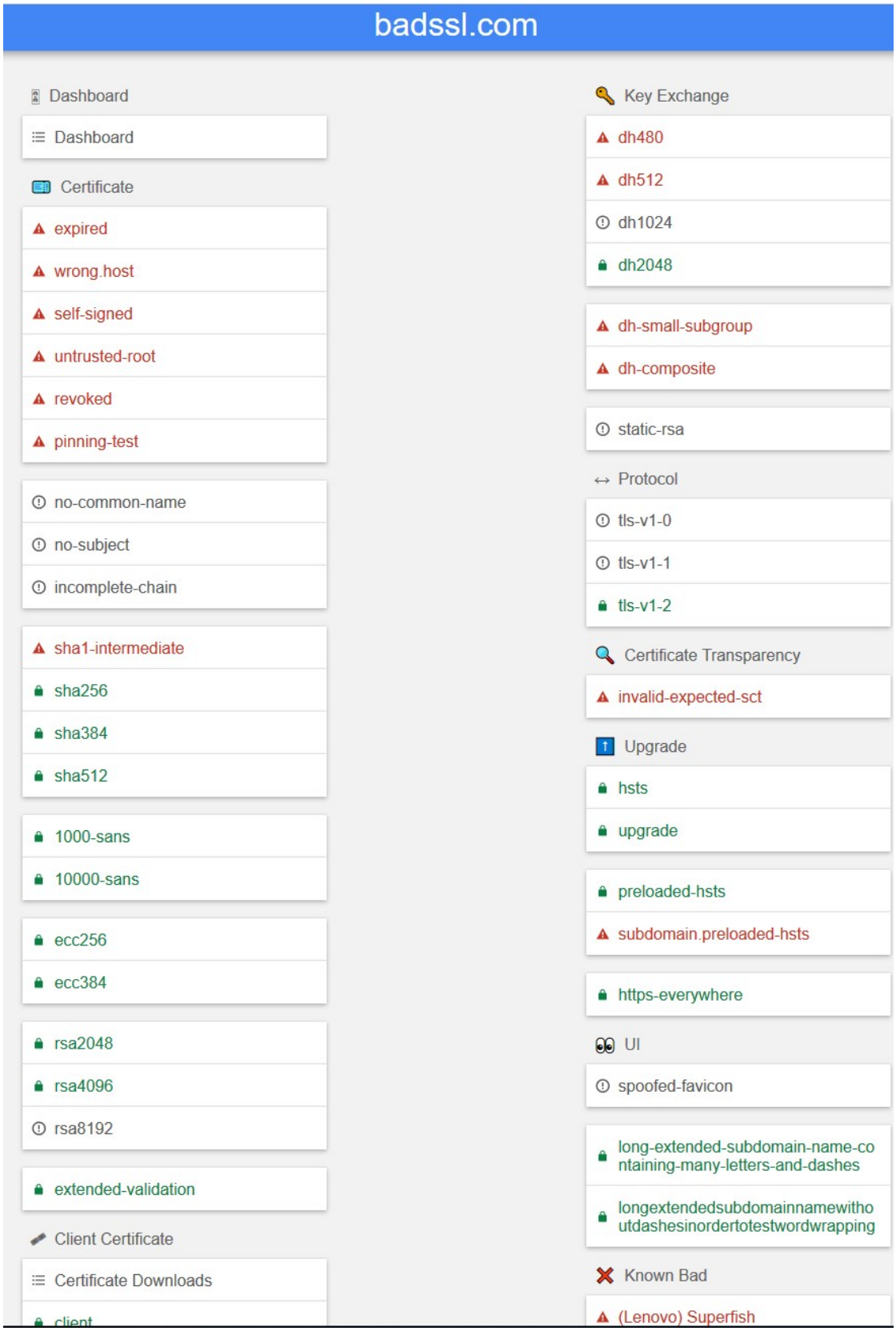
(PaloAlto Networks 2019c)

Appendix 11. Decryption profile Strict SSL control

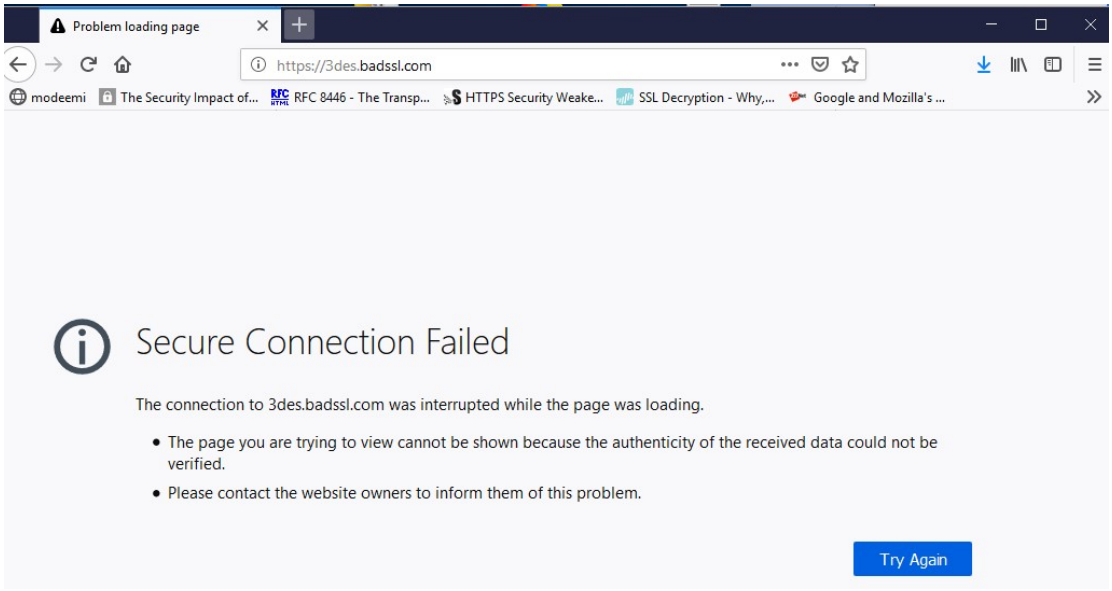




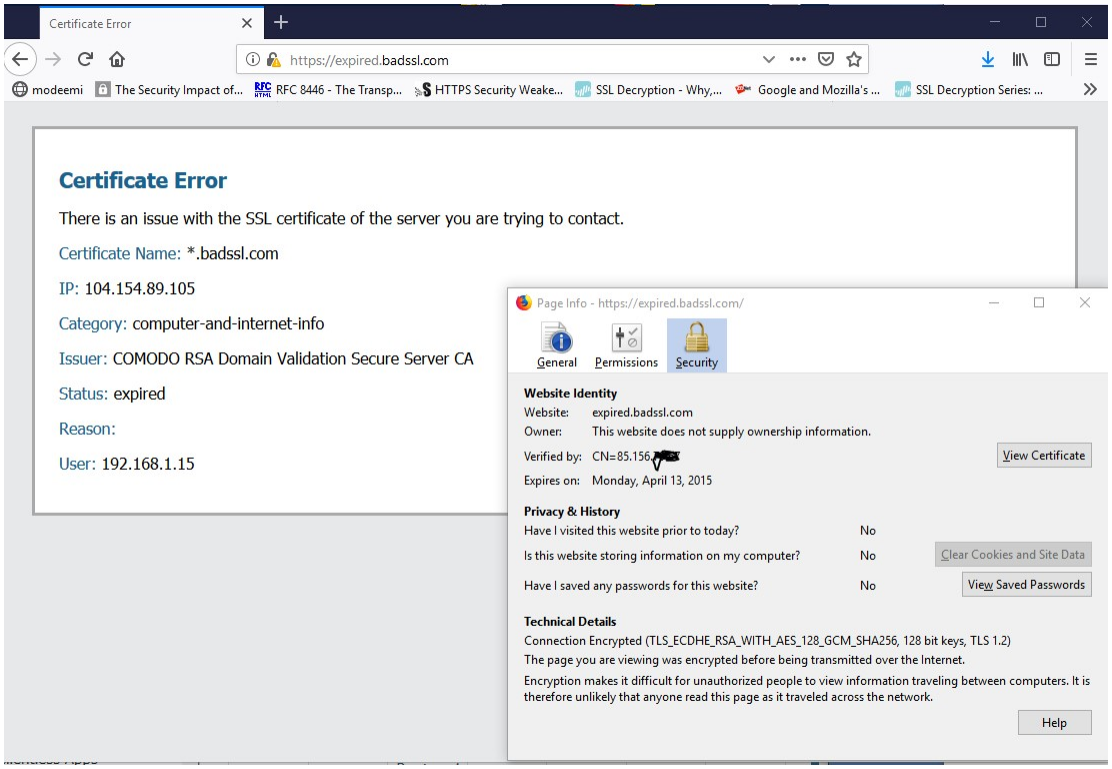
Appendix 12. Badssl.com test site for certificate testing



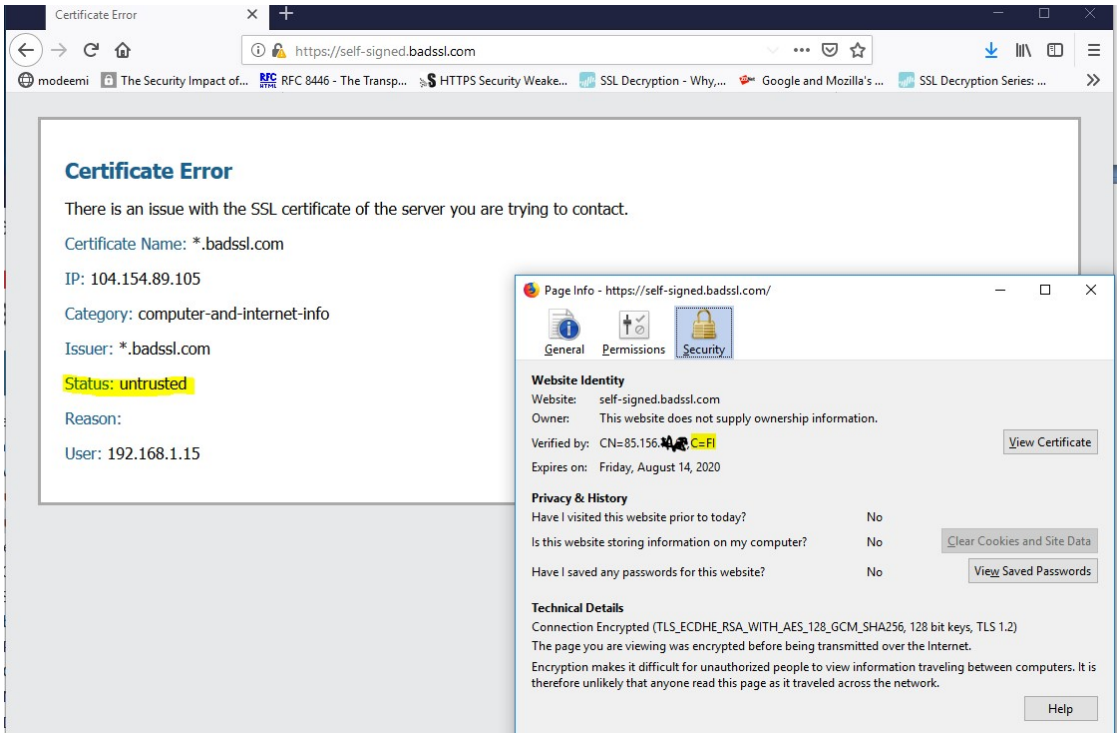
Appendix 13. 3DES cipher with strict SSL control profile

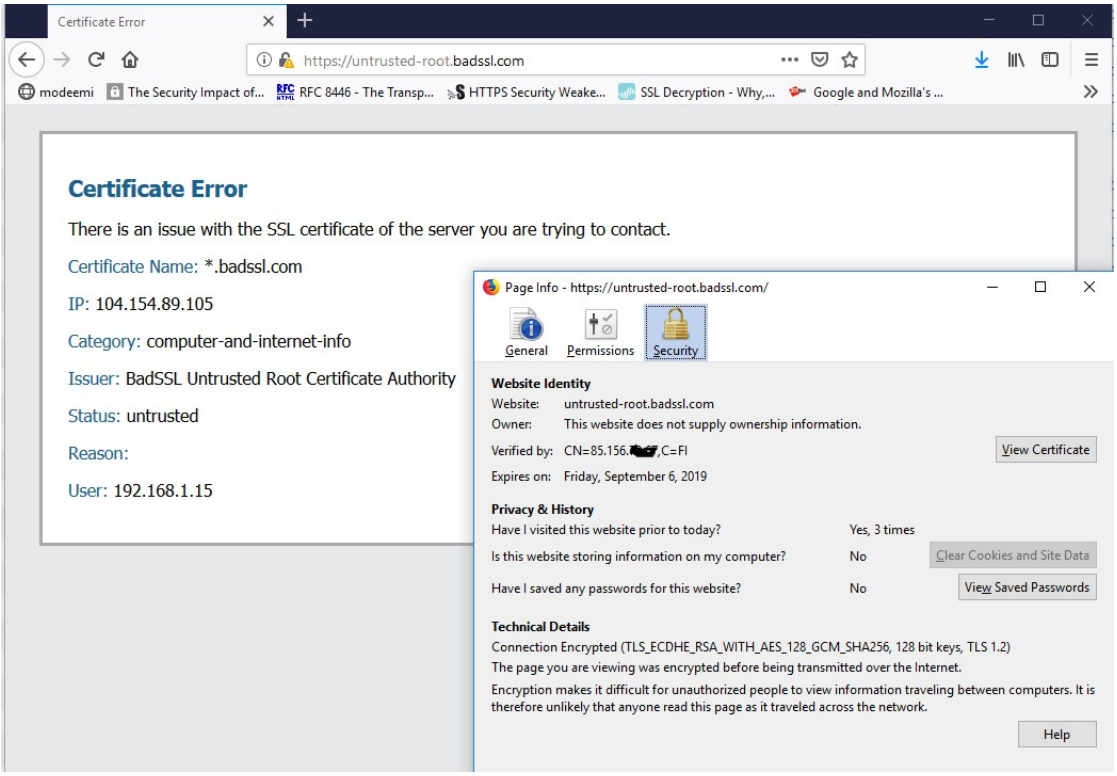


Appendix 14. Expired certificate after decryption

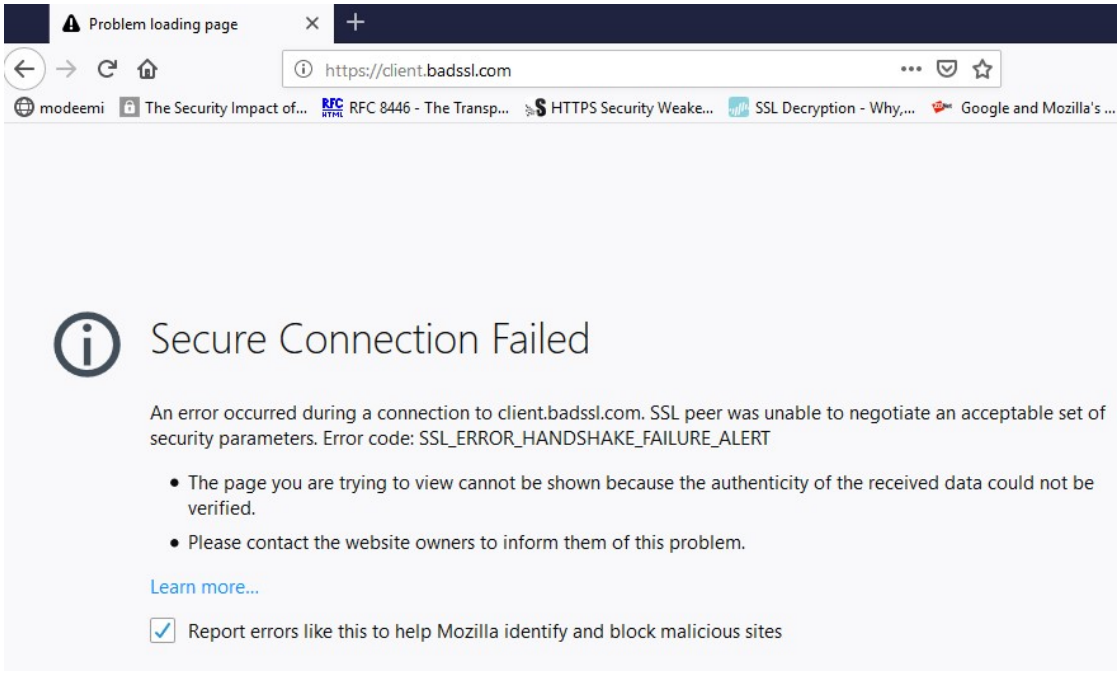


Appendix 15. Self-signed certificate and untrusted root certificate ends up using forward untrust certificate

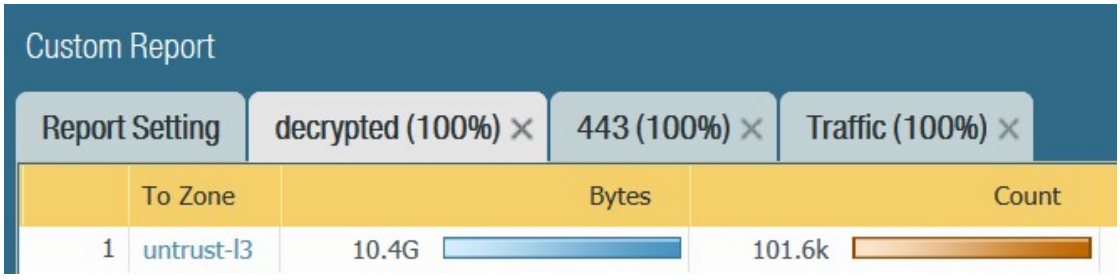


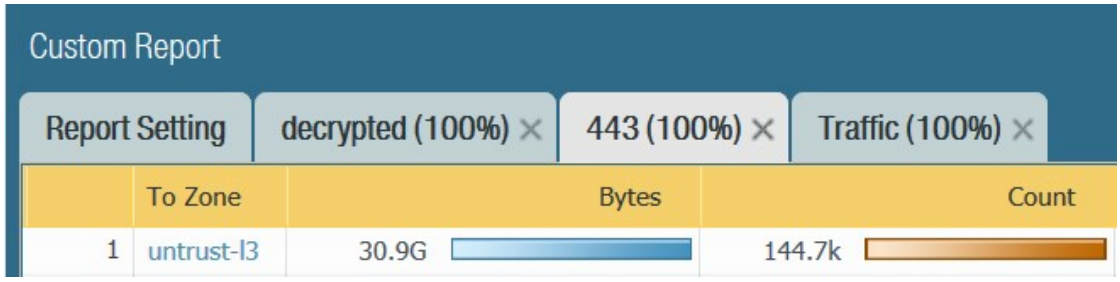


Appendix 16. Client certificate authentication with loose and strict



Appendix 17. Decrypted traffic amount compared to encrypted 7 days





Appendix 18. NGFW dataplane load during SSL decryption tests