TLS 1.3 : Solving new challenges for next-generation firewalls (NGFW)

Pass The Salt 2019
Who are we?

- **Nicolas Pamart**
  Apprentice Developer
  
  Does stuff,
  Did not have a choice to talk
  In front of you

  nicolas.pamart@stormshield.eu

- **Damien Deville**
  Technical Leader
  
  Does stuff

  damien.deville@stormshield.eu

- **Thomas Malherbe**
  Developer
  
  Does stuff

  thomas.malherbe@stormshield.eu

www.stormshield.com
We’re on the network ...
We protect users & enforce company policy
With our state of the art IPS

- TLS analysis
- Encrypted traffic detection
- Whitelist of ciphersuites
- Application filtering
- Vulnerability detection
Focus on TLS application filtering
TLS: Transport layer security

A TLS connection

HTTP or FTP, SMTP, ...
TLS
TCP or UDP
IP
Ethernet

TLS in the network stack
TLS 1.2 - Handshake

TLS 1.2 – Analysed Handshake

[ ] = Encrypted
With these data:

- SNI (e.g. ww.google.com)
- Server certificate
- Client certificate
- IP reputation

Application Policy

PASS

BLOCK
But now ... TLS 1.3 encrypts server certificate
Brand new TLS 1.3 handshake
We are passive*, we do not decrypt

*On the TLS layer
Server certificate is a public information
About certificates

User

SNI=www.stormshield.com

Common name=*.stormshield.com

Server

User

SNI=www.random.com

Common name=*.random.com
How-to: Get the same certificate

- Send the **same** server name indication (SNI)
- Propose the **same** cipherlist
- Send **our own** KeyShare extension
Wait, usually kernels don’t speak TLS!
Dear userspace daemon, talk for me
Yay ! We saved our feature
But for each connection ?!
OPTIMIZATION
Let’s cache certificate!

Pros 😊

• 0 delay certificate retrieval
• Less load on server
• Less load on NGFW

Cons 😞

• Design the cache: tune entry expiration date & cache size
• Design the cache #2: do something that works
Let’s cache it!
Let’s cache it!
How do we identify cache entries?

Entry key

- SNI
- Server's IP address

Certificate cache

- Certificate
- Certificate
- Certificate
- Certificate

Can be missing!
Handling session resumption
TLS 1.3 session resumption
TLS 1.3 session resumption : limitations

- Not really impacting our solution
  
  We base ourselves on **ClientHello information**
  
  => SNI is « theoretically* » provided in resumption ClientHello

- Some malicious peers could **not provide SNI** during resumption, thus breaking our filtering

*RFC 8446 section 4.2.11 : Pre-Shared Key Extension*
Simple, just check the presence of SNI no?
The problem with SNI

- SNI is not mandatory ...
- Need to check if original session was initiated with SNI
- How to do that?
Another cache ... for the SNI !
The big picture

1. Is resumption ClientHello?
   - Yes
   - No
     2. Is the certificate in cache?
        - Yes
          5. Is SNI coherent?
             - Yes
             - No
               3. Retrieve certificate
               4. Analyze certificate
        - No
          6. Is certificate valid?
             - Yes
               PASS
             - No
               BLOCK
SNI not coherent
SNI coherent & Cache HIT - PASS
SNI coherent & Cache HIT - BLOCK

Client

IPS (kernel)

Daemon (userspace)

Server

ClientHello

[PSK extension]

BUFFER ClientHello

SNI coherent

Certificate Cache HIT

IPS Analysis

EMPTY ClientHello buffer

Drop of ClientHello

(Crafted) TCP Reset
SNI coherent & Cache MISS - PASS
Proof of concept
PoC: design
PoC: supported features

- SNI coherence cache
- Certificate Caching
- Application blacklisting
  => Statistics gathering
PoC: results for 1 day / 1 user

Cache misses: 259
SNI incoherences: 0
Total: 2509
=> Ratio: 10.32% cache miss
Final note
World is safer now

- Facebook is blocked again
- That’s how we saved the world
20+ Open jobs!

Villeneuve d’Ascq – Paris (ILM) - Lyon

www.stormshield.com/join-us/
Thank you
Looking forward to hearing from you

Get in Touch

22, rue du Gouverneur Général Éboué
92130 Issy-les-Moulineaux FRANCE

+33 (0) 9 69 32 96 29

nicolas.pamart@stormshield.eu
damien.deville@stormshield.eu
thomas.malherbe@stormshield.eu
About encrypted SNI (eSNI)

- Currently as a draft
- Can break our solution as we are not able to obtain the SNI
- Encrypted via key given in DNS

=> **Solution:** We also analyze DNS traffic

(If you use DNSsec on top of that you may beat us)
TLS in kernel

• Requires to have the whole chain of cert in kernel

  => Not enough memory to do that, too costly

• It is technically possible to do TLS in kernel
About PSK-only servers (if it exists)

- PSK can be used to authenticate server
  => Thus no need for server certificate
  => Our solution don’t work (or don’t apply)

- **Solution**: Whitelist PSK-only servers
TLS 1.3 early-data

- Stripped when mimicking ClientHello
- Concerns about anti-replay
- We can’t provide sufficient security for anti-replay
TLS 1.2 handshake

[ ] = Encrypted
TLS 1.3 handshake
TLS 1.3 session resumption
TLS 1.3 0-RT == Resumption + Early data