pkt300 – defcon16

Challenge explained

Summary

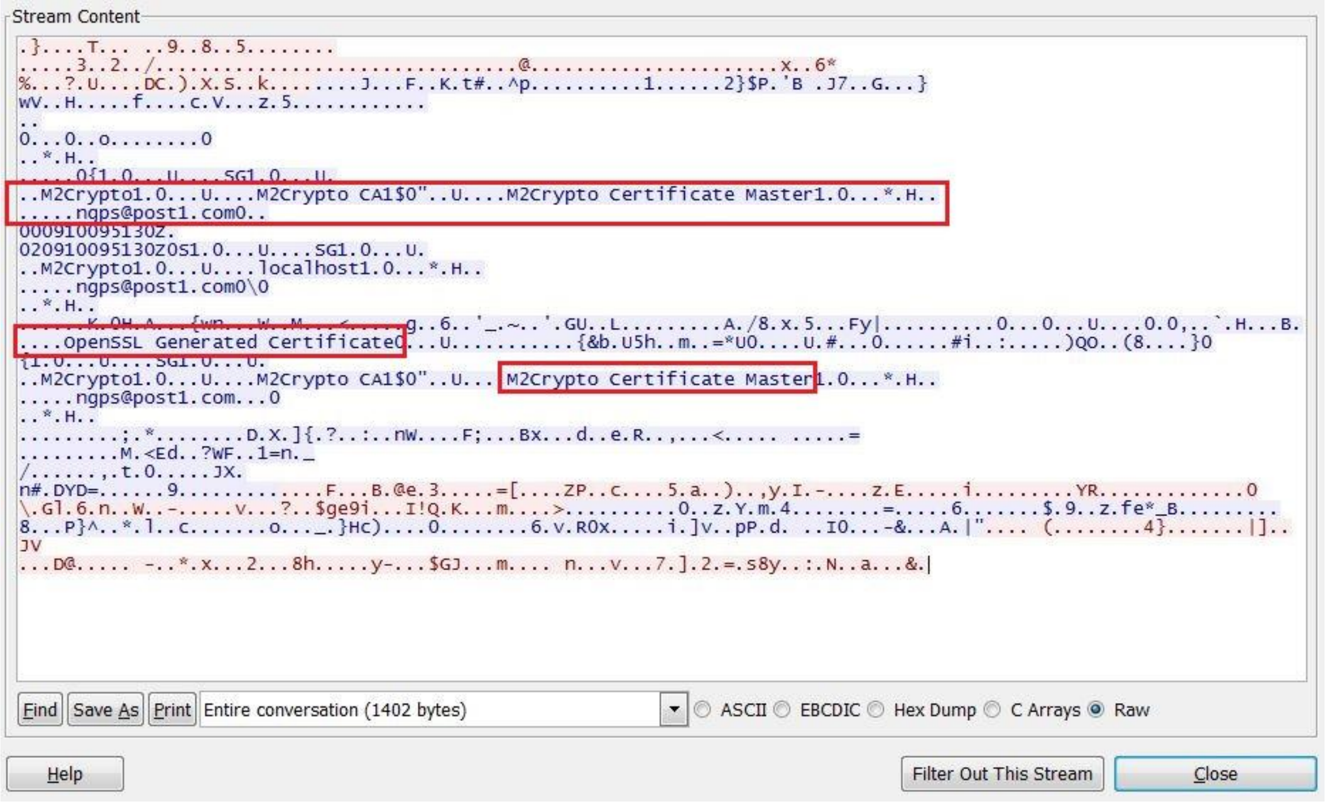
- Hint
- First clue
- Second clue
- SSL connection
- Third clue
- Final step

Hint

Google is your friend: *inurl:server.pem*

First clue

- Look at the packet stream with the « **Follow TCP stream** » feature



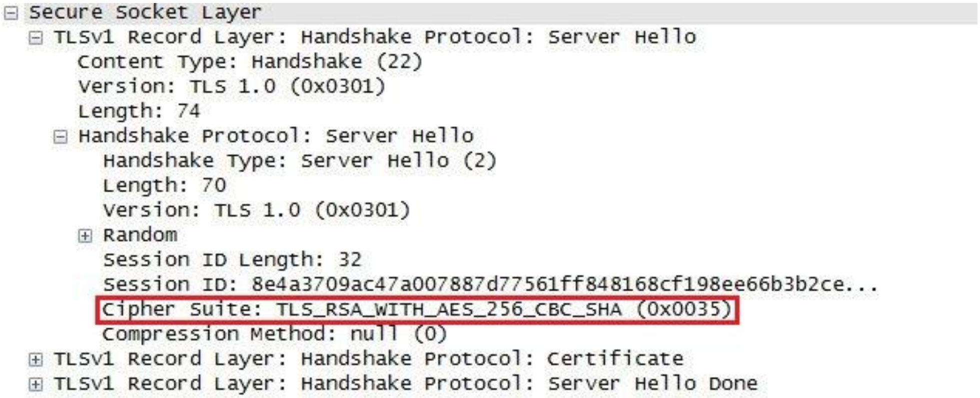
Stream Content

```
.}....T... ..9..8..5........
.....3..2../................................@......................x..6*
%...?.U....DC.).X.S..k........J...F..K.t#..^p..........1......2}$P.'B .J7..G...}
wV..H.....f....c.V...z.5...........
..
0...0..o........0
..*.H..
.....0{1.0...U....SG1.0...U.
..M2Crypto1.0...U....M2Crypto CA1$0"..U....M2Crypto Certificate Master1.0...*.H..
.....ngps@post1.com0..
000910095130Z.
020910095130Z0S1.0...U....SG1.0...U.
..M2Crypto1.0...U....localhost1.0...*.H..
.....ngps@post1.com0\0
..*.H..
.......K.OH.A...{wn...W..M...<......g..6..'_.~..'.GU..L.........A./8.x.5...Fy|..........0...0...U....0.0,..`.H...B.
....OpenSSL Generated Certificate0...U...........{&b.U5h..m..=*U0....U.#...0......#i..:.....)QO..(8....}0
{1.0...U....SG1.0...U.
..M2Crypto1.0...U....M2Crypto CA1$0"..U....M2Crypto Certificate Master1.0...*.H..
.....ngps@post1.com...0
..*.H..
.........;.*........D.X.]{.?..:..nW....F;...Bx...d..e.R..,...<..... .....=
.........M.<Ed..?WF..1=n._
/......,.t.0.....JX.
n#.DYD=......9.............F...B.@e.3.....=[....ZP..c....5.a..)..,y.I.-....z.E.....i.........YR.............0
\.Gl.6.n..W..-.....v...?..$ge9i...I!Q.K...m....>..........0..z.Y.m.4........=.....6.......$.9..z.fe*_B.........
8...P}^..*.l..c.......o..._.}Hc)....0........6.v.R0x.....i.]v..pP.d. ..I0...-&...A.|".... (........4}.......|]..
JV
...D@..... -..*.x...2...8h.....y-...$GJ...m.... n...v...7.].2.=.s8y..:.N..a...&.|
```

- We guess that the communication is protected with SSL

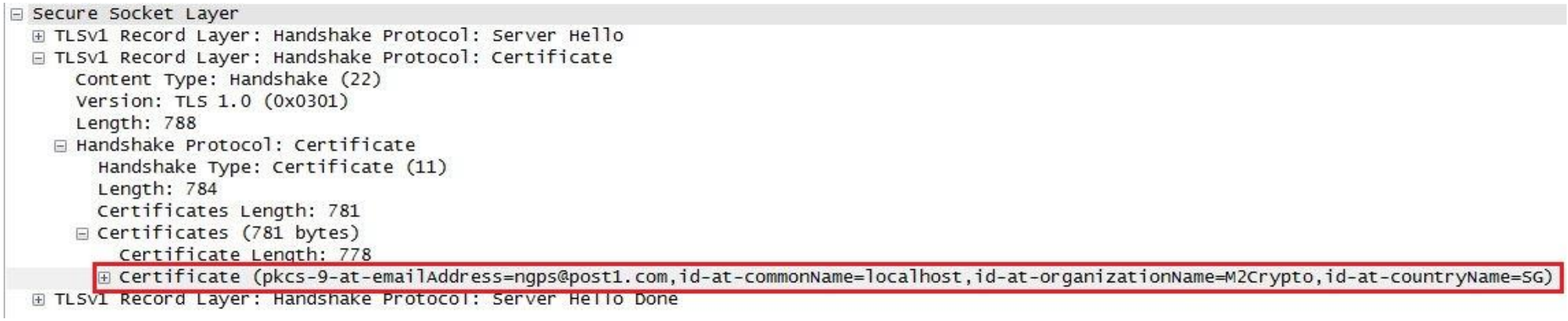
Second clue

- Use the **decode as** feature to have a deeper look into the SSL stream
- SSL options



```
⊟ Secure Socket Layer
  ⊟ TLSv1 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 74
    ⊟ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 70
        Version: TLS 1.0 (0x0301)
      ⊞ Random
        Session ID Length: 32
        Session ID: 8e4a3709ac47a007887d77561ff848168cf198ee66b3b2ce...
        Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
        Compression Method: null (0)
  ⊞ TLSv1 Record Layer: Handshake Protocol: Certificate
  ⊞ TLSv1 Record Layer: Handshake Protocol: Server Hello Done
```

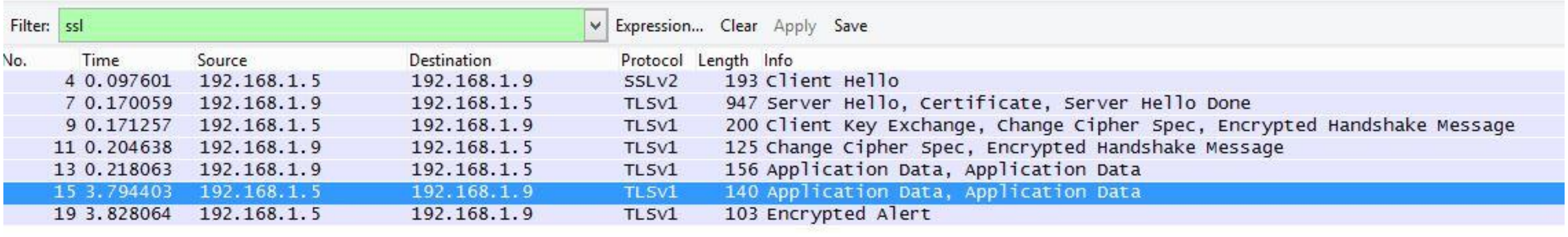
- Find the owner



```
⊟ Secure Socket Layer
  ⊞ TLSv1 Record Layer: Handshake Protocol: Server Hello
  ⊟ TLSv1 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 788
    ⊟ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 784
        Certificates Length: 781
      ⊟ Certificates (781 bytes)
          Certificate Length: 778
        ⊞ Certificate (pkcs-9-at-emailAddress=ngps@post1.com,id-at-commonName=localhost,id-at-organizationName=M2Crypto,id-at-countryName=SG)
  ⊞ TLSv1 Record Layer: Handshake Protocol: Server Hello Done
```

SSL connection

- In this .pcap you saw SSL messages
- They settle the SSL connection as follow



Filter: ssl

```
No.   Time      Source        Destination   Protocol Length Info
    4 0.097601  192.168.1.5   192.168.1.9   SSLv2     193 Client Hello
    7 0.170059  192.168.1.9   192.168.1.5   TLSv1     947 Server Hello, Certificate, Server Hello Done
    9 0.171257  192.168.1.5   192.168.1.9   TLSv1     200 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
   11 0.204638  192.168.1.9   192.168.1.5   TLSv1     125 Change Cipher Spec, Encrypted Handshake Message
   13 0.218063  192.168.1.9   192.168.1.5   TLSv1     156 Application Data, Application Data
   15 3.794403  192.168.1.5   192.168.1.9   TLSv1     140 Application Data, Application Data
   19 3.828064  192.168.1.5   192.168.1.9   TLSv1     103 Encrypted Alert
```

- TLS is described in RFC2246

- Next slide:

  Blue => client

  Violet => server

SSL connection

- Client hello: client wants to connect to a server
- Server hello: server responds to client
- Certificate: server sends its own certificate (x509)
- Server hello done: server indicates that its hello phase is finished
- Client key exchange: client sets the premaster key (RSA-encrypted secret)
- Change cipher spec: client indicates that following information will be encrypted
- Encrypted handshake message: handshake finished for the client
- Change cipher spec: server indicates that following information will be encrypted
- Encrypted handshake message: handshake finished for the server
- Application data: protected application data
- Encrypted alert: closing notification

- Interesting information here

Third clue

- What is a .PEM file

  Privacy Enhanced Mail Security Certificate is a container format that could contain a certificate, a public key and a private key.

- Find the .pem file using the given **google dork and the owner info**

- **Google dorks / hacks** are a set of expressions to perform advanced google searches.

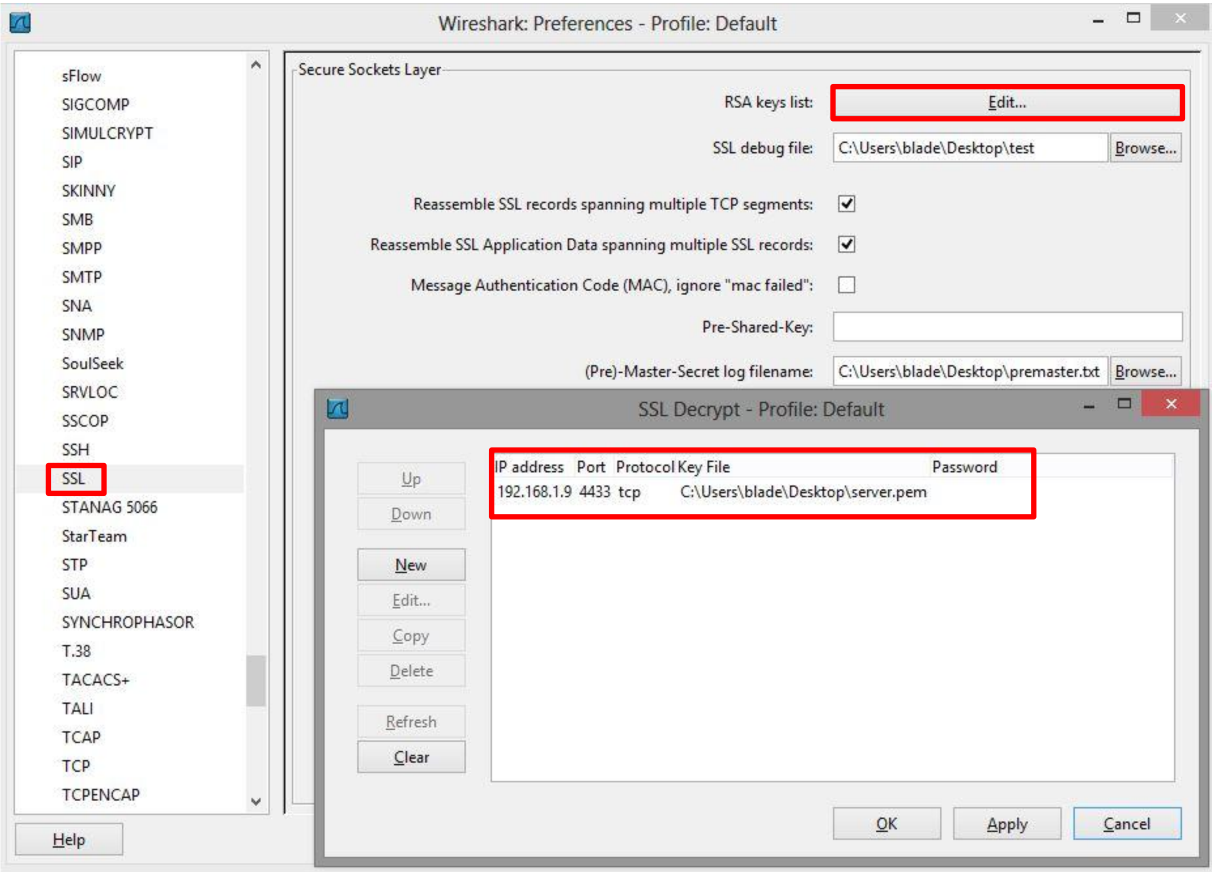
  e.g.: inurl=server.pem will look for the string server.pem inside the URL.

- Here you can try:

  m2crypto inurl:server.pem

Decryption

- Use the private key to decrypt the communication (inside .pem) (Edit>Preferences>Protocols>SSL)



Wireshark: Preferences - Profile: Default

Secure Sockets Layer

RSA keys list: Edit...

SSL debug file: C:\Users\blade\Desktop\test

Reassemble SSL records spanning multiple TCP segments: ☑

Reassemble SSL Application Data spanning multiple SSL records: ☑

Message Authentication Code (MAC), ignore "mac failed": □

Pre-Shared-Key:

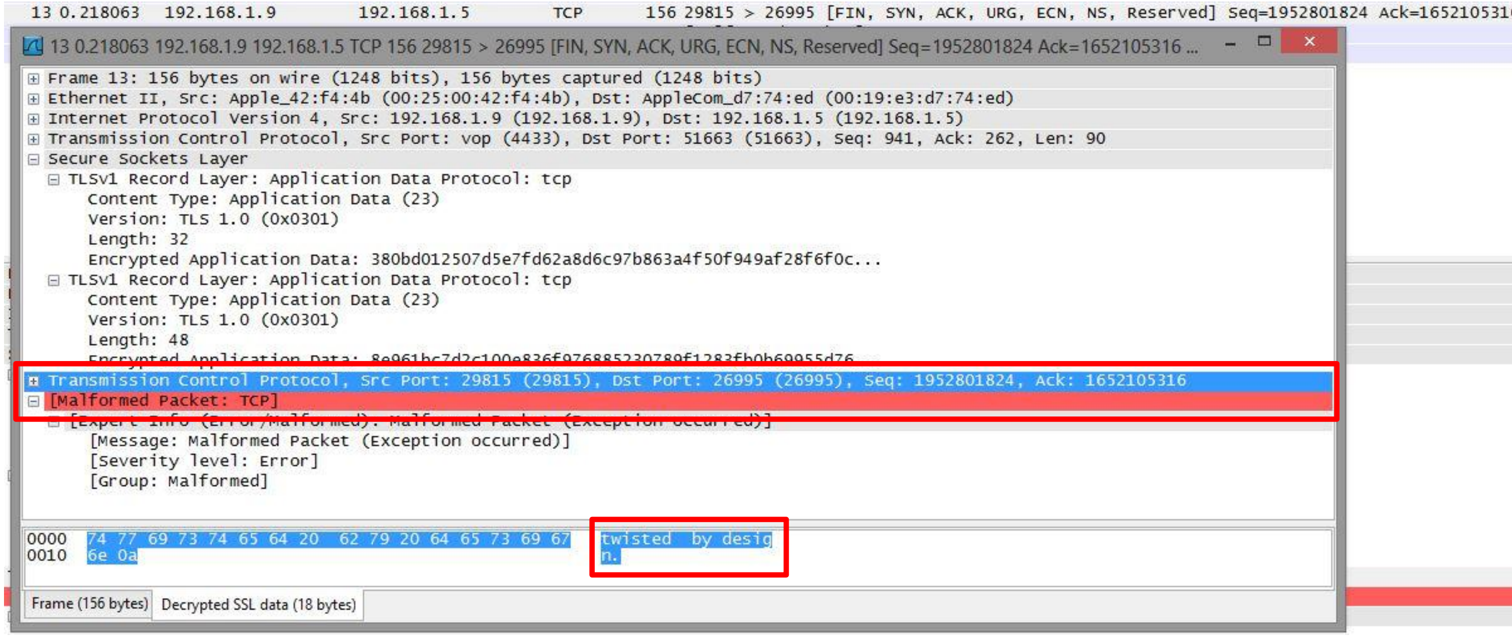
(Pre)-Master-Secret log filename: C:\Users\blade\Desktop\premaster.txt

SSL Decrypt - Profile: Default

IP address	Port	Protocol	Key File	Password
192.168.1.9	4433	tcp	C:\Users\blade\Desktop\server.pem	

Final step

- Spot the malformed packet
- Examine it closely



```
13 0.218063   192.168.1.9        192.168.1.5        TCP      156 29815 > 26995 [FIN, SYN, ACK, URG, ECN, NS, Reserved] Seq=1952801824 Ack=1652105316 Win=26983[Malformed Pac
```

13 0.218063 192.168.1.9 192.168.1.5 TCP 156 29815 > 26995 [FIN, SYN, ACK, URG, ECN, NS, Reserved] Seq=1952801824 Ack=1652105316 ...

```
⊞ Frame 13: 156 bytes on wire (1248 bits), 156 bytes captured (1248 bits)
⊞ Ethernet II, Src: Apple_42:f4:4b (00:25:00:42:f4:4b), Dst: AppleCom_d7:74:ed (00:19:e3:d7:74:ed)
⊞ Internet Protocol Version 4, Src: 192.168.1.9 (192.168.1.9), Dst: 192.168.1.5 (192.168.1.5)
⊞ Transmission Control Protocol, Src Port: vop (4433), Dst Port: 51663 (51663), Seq: 941, Ack: 262, Len: 90
⊟ Secure Sockets Layer
  ⊟ TLSv1 Record Layer: Application Data Protocol: tcp
      Content Type: Application Data (23)
      Version: TLS 1.0 (0x0301)
      Length: 32
      Encrypted Application Data: 380bd012507d5e7fd62a8d6c97b863a4f50f949af28f6f0c...
  ⊟ TLSv1 Record Layer: Application Data Protocol: tcp
      Content Type: Application Data (23)
      Version: TLS 1.0 (0x0301)
      Length: 48
      Encrypted Application Data: 8e961bc7d2c100e836f976885230789f1283fb0b69955d76...
⊞ Transmission Control Protocol, Src Port: 29815 (29815), Dst Port: 26995 (26995), Seq: 1952801824, Ack: 1652105316
⊟ [Malformed Packet: TCP]
  ⊟ [Expert Info (Error/Malformed): Malformed Packet (Exception occurred)]
      [Message: Malformed Packet (Exception occurred)]
      [Severity level: Error]
      [Group: Malformed]

0000   74 77 69 73 74 65 64 20  62 79 20 64 65 73 69 67   twisted  by desig
0010   6e 0a                                              n.
```

Questions ?

?

Links

- PEM
- http://tools.ietf.org/html/rfc1421

- Wireshark
- http://www.wireshark.org/

- Challenge back in 2008 (Spanish)
- http://dumacx.blogspot.fr/2010/05/el-fin-de-semana-pasado-entre-el-21-y.html

- Challenge files
- http://stalkr.net/files/defcon/18/quals/packet300/

- Google hacks
- http://it.toolbox.com/blogs/managing-infosec/google-hacking-master-list-28302
- http://www.hackersforcharity.org/ghdb/

Thank you