Snoopcon 2017

Binary Protocols: Analysis & Interception

Jérémy Brun

French BT pentest team

Snoopcon 2017

Agenda

1. Introducing the tool CANAPE
2. STARTTLS-like traffic
3. Compressed traffic
4. .NET Binary format
5. Java Serialized data

Snoopcon 2017

1. Introducing the tool CANAPE

Thick client auditing

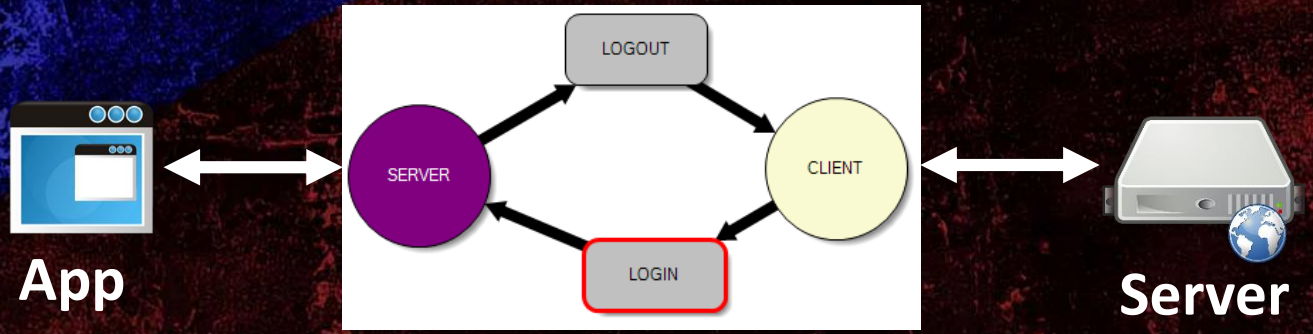
- Network traffic analysis is important
- Tools usually recommended for binary protocols: Tcpdump, Wireshark…
- Problems:
  - Not well adapted for encrypted or serialized data (need for post-processing with custom tools);
  - No on-the-fly interception with live edition possibility.
- We need a tool like “Burp” for binary protocols

Snoopcon 2017

Snoopcon 2017

CANAPE FTW

- Network testing tool, dev by James Forshaw
- Act as SOCKS proxy between client/server
- Config via graphs, each node represents an operation
- Scripting support: C#, Python



Snoopcon 2017

2. STARTTLS-like traffic

WTF ?

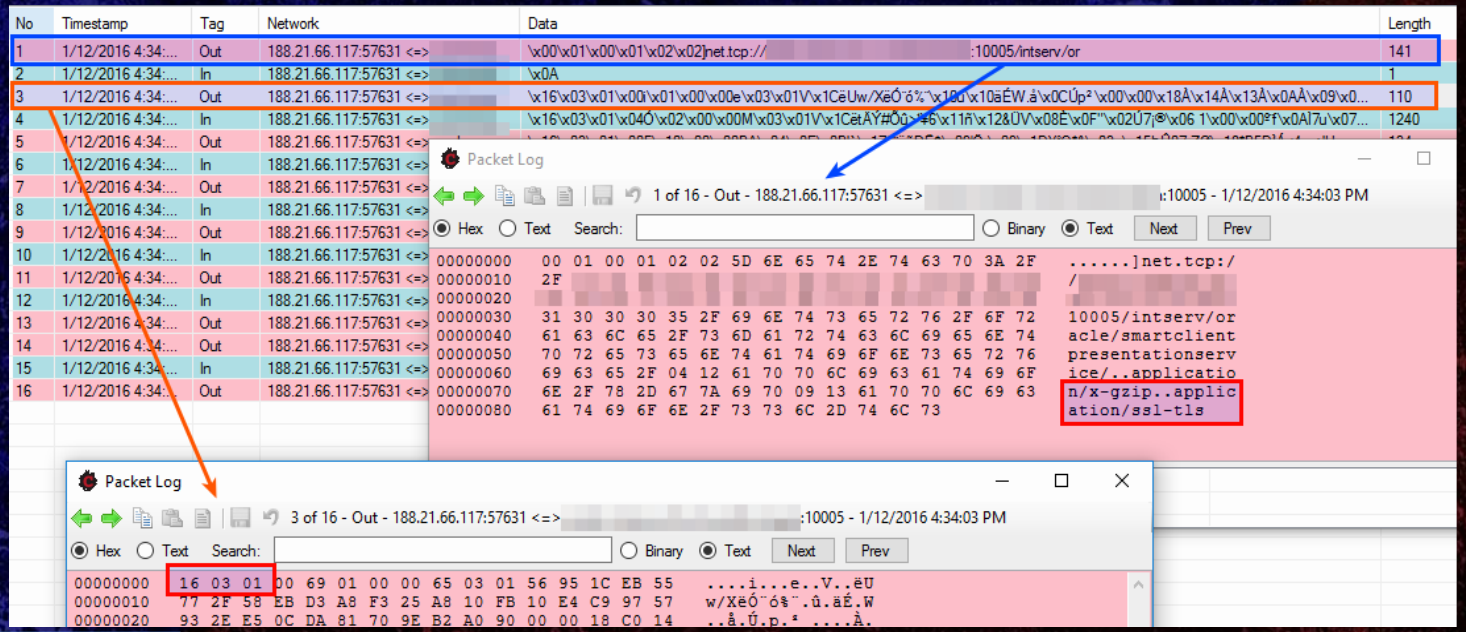
No	Timestamp	Tag	Network	Data	Length
1	1/12/2016 4:34:...	Out	188.21.66.117:57631 <=>	\x00\x01\x00\x01\x02\x02]net.tcp://_:10005/intserv/or	141
2	1/12/2016 4:34:...	In	188.21.66.117:57631 <=>	\x0A	1
3	1/12/2016 4:34:...	Out	188.21.66.117:57631 <=>	\x16\x03\x01\x00i\x01\x00\x00e\x03\x01V\x1CëUw/XëÓ¨ó%^\x10û\x10äÉW.à\x0CÚp²\x00\x00\x18Å\x14Å\x13Å\x0AÅ\x0...	110
4	1/12/2016 4:34:...	In	188.21.66.117:57631 <=>	\x16\x03\x01\x04Ó\x02\x00\x00M\x03\x01V\x1CëtÄÝ#Ðû>¹¥6\x11ñ\x12&ÜV\x08È\x0F'"\x02Ú7¡®\x06 1\x00\x00ªf\x0Al7u...	1240
5	1/12/2016 4:34:...	Out	188.21.66.117:57631 <=>	\x16\x03\x01\x00F\x10\x00\x00BA\x04\x0E\x0B'è\x17£¨öðÐÉ$\x09jÓ-\x09\x1D¥°Q*ã\x03g\x15Þ Û07 ZØ\x18*B5DlÁ<4zz¿'H	134
6	1/12/2016 4:34:...	In	188.21.66.117:57631 <=>	\x14\x03\x01\x00\x01\x01\x16\x03\x01\x000\x1AÄ\x11ýÞ¿ó¡¼«\x00t¹§v=\x08H\x18\x15Ê/ëbH\x1B¶¶µ{2× ".\x0BÉ°Ôt[	59
7	1/12/2016 4:34:...	Out	188.21.66.117:57631 <=>	\x17\x03\x01\x00 ¢Ð¼4n¦4\x004û²+ÛiãxK³4\x02\x0D\x06ÉÃ®y»ç\x00_\x10	37
8	1/12/2016 4:34:...	In	188.21.66.117:57631 <=>	\x17\x03\x01\x00 Büc{>\x0AØ.óx«³71íØV¥d®\x1B+¡ÔL\x01\x19ã]	37
9	1/12/2016 4:34:...	Out	188.21.66.117:57631 <=>	\x17\x03\x01\x02åm½µÉÐ()·Væ¥c-\x05?@±638õ\x11}½è{aÐ¤4¾üï«êm&°ræðXÔ	661
10	1/12/2016 4:34:...	In	188.21.66.117:57631 <=>	\x17\x03\x01\x02'5÷È$Hûç\x1A_Ctiß6\x18DXèTµxÔ°%É\x17ÃÜ'b\x00åy\x14o¢2A\x0C"\x02'ÍN.\x0F\x10\x04^	613
11	1/12/2016 4:34:...	Out	188.21.66.117:57631 <=>	\x17\x03\x01\x00 JôOt-\x13UÛàÖ\x10Mæ0;\x12p$hqµ§4ÕmEzàñ	37
12	1/12/2016 4:34:...	In	188.21.66.117:57631 <=>	\x17\x03\x01\x00 S\x03®\x0Eäö\x13'écT4!yé«ÇéÎ<GðlùH_üH	37
13	1/12/2016 4:34:...	Out	188.21.66.117:57631 <=>	\x00\x01\x00\x01\x02\x02]net.tcp://_:10005/intserv/or	120
14	1/12/2016 4:34:...	Out	188.21.66.117:57631 <=>	\x09\x13application/ssl-tls	21
15	1/12/2016 4:34:...	In	188.21.66.117:57631 <=>	\x0A	1
16	1/12/2016 4:34:...	Out	188.21.66.117:57631 <=>	\x16\x03\x01\x00i\x01\x00\x00e\x03\x01V\x1Cët-C7\x0AÉ·=S\x0DÌZ }~ÛÈH.¨qÅq^·ª\x00\x00\x18Å\x14Å\x13Å\x0AÅ\x09\x...	110

Cleartext + Encrypted traffic

=> STARTTLS is a way to turn a cleartext connection into an encrypted one

Snoopcon 2017

Modeling 2 States



State # 1
[Cleartext]

[TLS session opening]

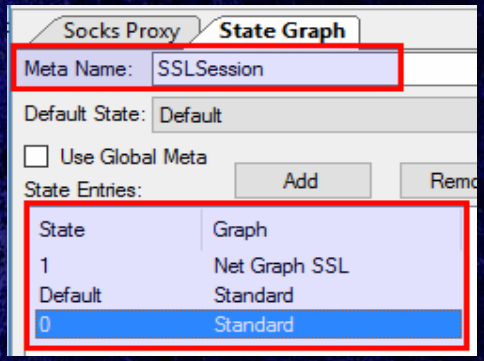
State #2
[Encrypted]

[TLS session closing]

Snoopcon 2017

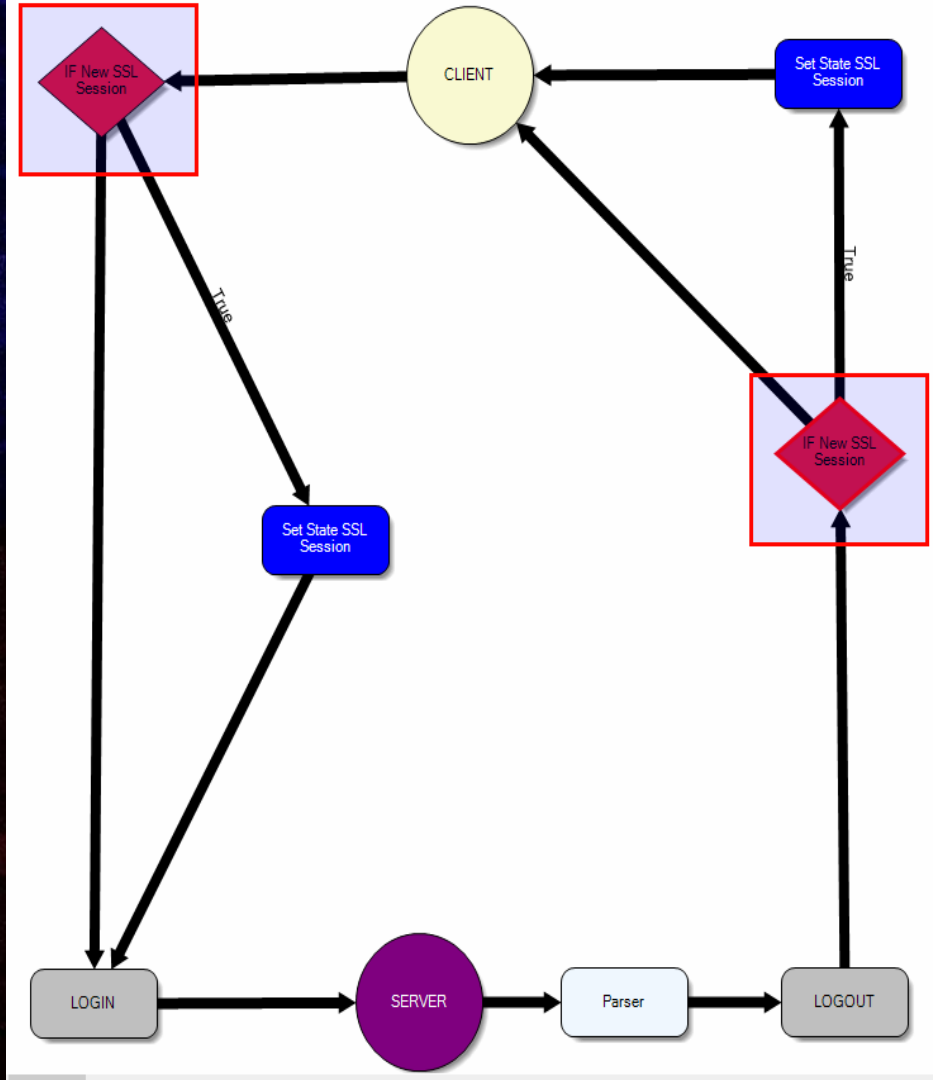
State #1: Cleartext

- Use the feature: State Graph



=> The graph to use will depends on the value of *SSLSession* var.

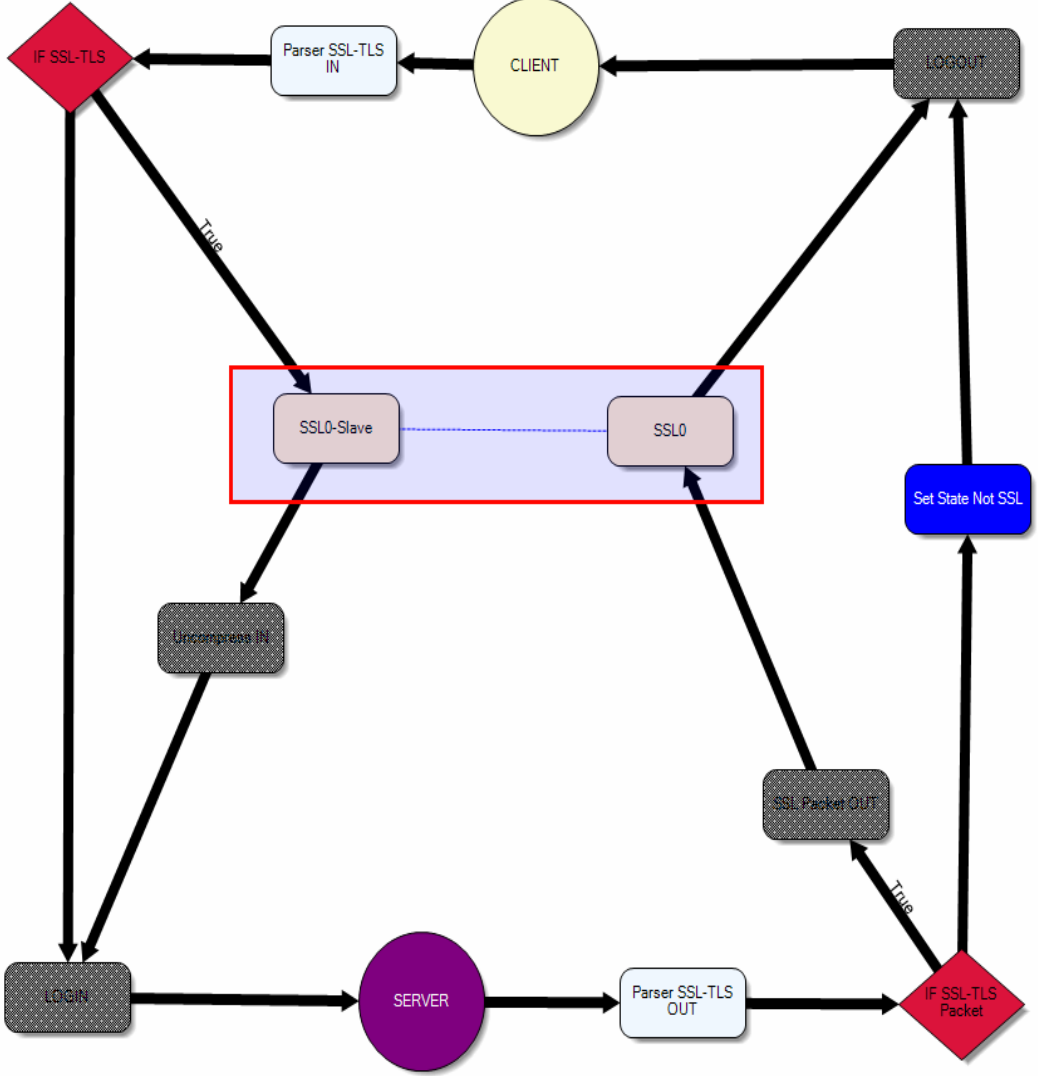
- Red node = Detect new TLS session
- Blue node = Change var value



Snoopcon 2017

State #2: Encrypted

- SSL/TLS decryption performed by linked-nodes “SSL Layer Section”
- If new cleartext data detected => change *SSLSession* variable value and go back to State #1 graph



Snoopcon 2017

Snoopcon 2017

3. Compressed traffic

Compression formats

- Deflate (RFC 1951)
- Zlib (RFC 1950) & Gzip (RFC 1952)

Both add header + footer to data compressed using Deflate algorithm

Tool “Compression Identifier”

- Detection
- Compression / Decompression

- Code can re-used inside CANAPE to decompress on-the-fly

```
[jbr:...ools/Compression_identifier]$ python compress_id.py -f ../../Samples/Zl

================================================================================
                        -- Compression Identifier --
================================================================================
  Supported compression formats:
  - Zlib (RFC 1950)
  - Deflate (RFC 1951)
  - Gzip (RFC 1952)


Automatic detection of compressed data

[~] Scan input and search for compression at the different offsets...
[+] Zlib compressed data found at offset 10

[~] Header: 78 9c
[~] +--- CMF = 78
[~] |  +--- CM (Compression Method)  = 8 - Deflate
[~] |  +--- CINFO (Compression Info) = 7
[~] +--- FLG = 9c
[~] |  +--- FCHECK = 28
[~] |  +--- FDICT  = 0 - No DICTID field
[~] |  +--- FLEVEL = 2 - Default algorithm

[~] Full hexdump of decompressed data:
0000  44 4d 45 53 53 41 47 45 5f 54 59 50 45 00 52 45   DMESSAGE_TYPE.RE
0010  50 5f 4c 4f 47 4f 4e 5f 4b 45 59 00 43 4d 46 5f   P_LOGON_KEY.CMF_
0020  56 45 52 53 49 4f 4e 00 00 00 00 02 61 43 48 41   VERSION.....aCHA
0030  4c 4c 45 4e 47 45 5f 53 54 52 00 00 20 a1 1c ac   LLENGE_STR.. ...
0040  fe e0 62 bf 6a 72 bc 45 c3 56 cb 97 41 30 32 59   ..b.jr.E.V..A02Y
0050  6b c3 ce 06 16 6a 14 a5 4e 59 4a ed 80 64 55 53   k....j..NYJ..dUS
0060  45 52 5f 44 45 54 41 49 4c 5f 46 49 45 4c 44 53   ER_DETAIL_FIELDS
0070  00 00 0b 57 4f 52 4b 42 4f 4f 4b 5f 4e 41 4d 45   ...WORKBOOK_NAME
0080  00 57 4f 52 4b 42 4f 4f 4b 5f 54 49 4d 45 53 54   .WORKBOOK_TIMEST
0090  41 4d 50 00 4f 52 47 41 4e 49 53 41 54 49 4f 4e   AMP.ORGANISATION
00a0  00 41 50 50 4c 49 43 41 54 49 4f 4e 00 57 54 53   .APPLICATION.WTS
```

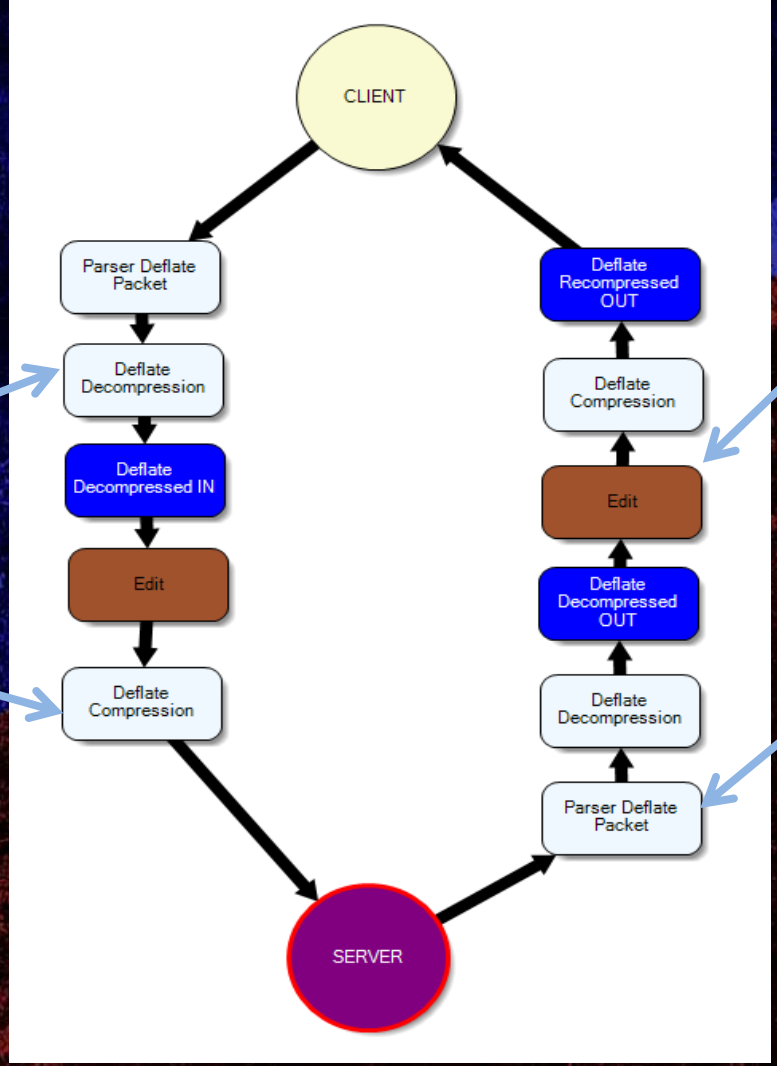
Snoopcon 2017

On-the-fly interception

Snoopcon 2017

Python/C# scripts implemented into Canape

Blue nodes = Packet Loggers



Node for on-the-fly edition

Compressed data may be splitted into several TCP packets. Parser will aggregate them.

Snoopcon 2017

4. .NET Binary format (WCF in netTcpBinding mode)

Specifications

- Serialize XML into binary
- **Standard format - [MC-NBFS]:**
  - *DictionaryString* structure: Most commonly used strings in SOAP (*Envelope, Header…*) are represented using even numbers (0x00, 0x02…)
- **Extended format: [MC-NBFSE]:**
  - Add a *StringTable*: new strings can be indexed using odd numbers (0x01, 0x03…).

String to encode	Encoded bytes (hex)
<s:Envelope	56 **02**
xmlns:a="http://www.w3.org/2005/08/addressing"	0B 01 61 **06**
xmlns:s="http://www.w3.org/2003/05/soap-envelope">	0B 01 73 **04**
<s:Header>	56 **08**
<a:Action	44 **0A**
s:mustUnderstand="1">	1E **00** 82
action</a:Action>	99 06 61 63 74 69 6F 6E
</s:Header>	01
<s:Body>	56 **0E**
<Inventory>	40 09 49 6E 76 65 6E 74 6F 72 79
0</Inventory>	81
</s:Body>	01
</s:Envelope>	01

Full specs:
https://msdn.microsoft.com/en-us/library/cc219190.aspx

Snoopcon 2017

Snoopcon 2017

Tool

Example

[MC-NBFSE]

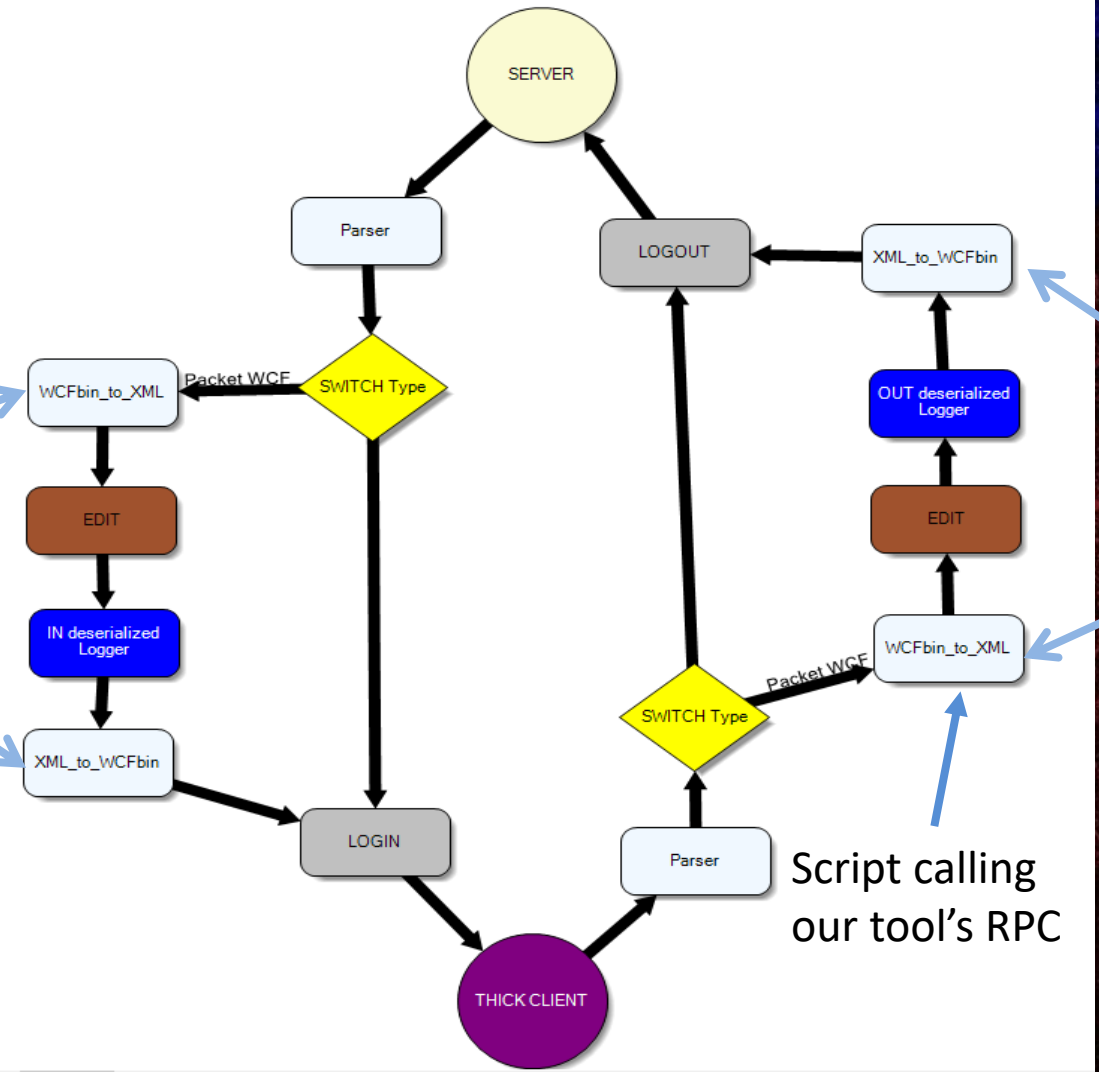
```
[jbr:...s/DotNetBInary_Deserializer]$ python netbinary_deserializer.py -f ../../Samples/NetBinary-MC-NBFSE/Samples_ChatApplication/chat_wcf_packet_01.raw --bin2xml --offset=3

================================================================================
                -- .NET Binary Serializer/Deserializer --
================================================================================
  Supported encoding formats:
  - [MC-NBFS]  - Standard
  - [MC-NBFSE] - With in-band dictionary (StringTable)
  (Used by WCF in netTcpBinding mode)


.NET Binary --> XML

[~] Input file length: 352 (0x160) bytes
[~] Scanning input for .NET Binary data and try to decode it...
[+] .NET Binary data - Format [MC-NBFSE] (with in-band dictionary) - decoded with success from offset 0x3
[~] In-band dictionary (partial StringTable):
[~] .   offsets= 0x3-0xE6 | len= 0xE3 bytes
[~] .   [0x01] http://schemas.microsoft.com/net/2006/05/peer/resolver/Resolve
[~] .   [0x03] net.tcp://192.168.142.191/ChatServer
[~] .   [0x05] Resolve
[~] .   [0x07] http://schemas.microsoft.com/net/2006/05/peer
[~] .   [0x09] http://www.w3.org/2001/XMLSchema-instance
[~] .   [0x0B] ClientId
[~] .   [0x0D] MaxAddresses
[~] .   [0x0F] MeshId

--------------------------------------------------------------------------------
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:a="http://www.w3.org/2005/08/addressing">
 <s:Header >
  <a:Action s:mustUnderstand="1">[[http://schemas.microsoft.com/net/2006/05/peer/resolver/Resolve|ST_0x01]]</a:Action>
  <a:MessageID >urn:uuid:c287b386-2c1f-458f-a3c3-abb8f1f30cf3</a:MessageID>
  <a:ReplyTo >
   <a:Address >http://www.w3.org/2005/08/addressing/anonymous</a:Address>
  </a:ReplyTo>
  <a:To s:mustUnderstand="1">[[net.tcp://192.168.142.191/ChatServer|ST_0x03]]</a:To>
 </s:Header>
 <s:Body >
  <[[Resolve|ST_0x05]] xmlns="[[http://schemas.microsoft.com/net/2006/05/peer|ST_0x07]]" xmlns:i="[[http://www.w3.org/2001/XMLSchema-instance|ST_0x09]]">
   <[[ClientId|ST_0x0b]] >7462646d-bfbb-421d-962d-e76bb7a16575</[[ClientId|ST_0x0b]]>
   <[[MaxAddresses|ST_0x0d]] >3</[[MaxAddresses|ST_0x0d]]>
   <[[MeshId|ST_0x0f]] >chatmesh</[[MeshId|ST_0x0f]]>
  </[[Resolve|ST_0x05]]>
 </s:Body>
</s:Envelope>
--------------------------------------------------------------------------------
```

StringTable rebuilt

Deserialized data

Snoopcon 2017

On-the-fly edition



Re-serialize

Deserialize

Script calling our tool's RPC

Snoopcon 2017

5. Java Binary Serialized

Specifications

- Java Objects implementing *Java.io.Serializable* interface can be serialized (converted into sequence of bytes).

- First bytes of Java Serialized data: **AC ED 00 05**

- When analyzing Java serialized data: Keep in mind that it can mix several objects and/or data of primitive types (Byte, Boolean, Char, Int, Float…)

Full specs:
https://docs.oracle.com/javase/8/docs/platform/serialization/spec/protocol.html

Snoopcon 2017

Tool

- Tool developed to work together with CANAPE
- Can be used for on-the-fly deserilization -> edition -> re-serialization
- Needs client's JAR in classpath.

```
================================================================================
                    -- Java Serializer/Deserializer --
================================================================================

Java Deserialization

[~] Input file length: 200 (0xc8) bytes

0000  ac ed 00 05 73 72 00 0a 54 63 70 50 61 79 6c 6f   ....sr..TcpPaylo
0010  61 64 ff 4e 16 c9 0f 64 db bd 02 00 07 43 00 05   ad.N...d.....C..
0020  63 68 61 72 31 44 00 07 64 6f 75 62 6c 65 31 46   char1D..double1F
0030  00 06 66 6c 6f 61 74 31 49 00 04 69 6e 74 31 4a   ..float1I..int1J
0040  00 05 6c 6f 6e 67 31 53 00 06 73 68 6f 72 74 31   ..long1S..short1
0050  4c 00 04 73 74 72 31 74 00 12 4c 6a 61 76 61 2f   L..str1t..Ljava/
0060  6c 61 6e 67 2f 53 74 72 69 6e 67 3b 78 70 00 78   lang/String;xp.x
0070  40 4b 87 0a 3d 70 a3 d7 c2 b4 19 9a 00 00 00 7b   @K..=p.........{
0080  ff ff ff ff fe 93 60 a3 00 3b 74 00 16 49 20 61   ......`..;t..I a
0090  6d 20 61 20 53 74 72 69 6e 67 20 70 61 79 6c 6f   m a String paylo
00a0  61 64 2e 77 23 41 42 43 44 05 01 64 00 63 40 4b   ad.w#ABCD..d.c@K
00b0  87 0a 3d 70 a3 d7 c2 b4 19 9a 00 00 7a 69 ff ff   ..=p........zi..
00c0  ff ff fe 93 60 a3 05 39                           ....`..9

[~] Scanning input for Java Serialized data and try to deserialize it...
[+] Java Serialized data header found at offset 0x0


--------------------------------------------------------------------------------
[~] [0x00] Java Object (converted into XML):
<TcpPayload>
  <int1>123</int1>
  <float1>-90.05</float1>
  <double1>55.055</double1>
  <short1>59</short1>
  <str1>I am a String payload.</str1>
  <long1>-23895901</long1>
  <char1>x</char1>
</TcpPayload>

--------------------------------------------------------------------------------
[~] [0x01] Block raw data - length = 35 (0x23) bytes:
0000  41 42 43 44 05 01 64 00 63 40 4b 87 0a 3d 70 a3   ABCD..d.c@K..=p.
0010  d7 c2 b4 19 9a 00 00 7a 69 ff ff ff ff fe 93 60   .......zi......`
0020  a3 05 39                                          ..9
```

Deserialized Java Object

Raw data

Snoopcon 2017

Snoopcon 2017

Conclusion

Conclusion

- Python tools available on my github:

  https://github.com/koutto

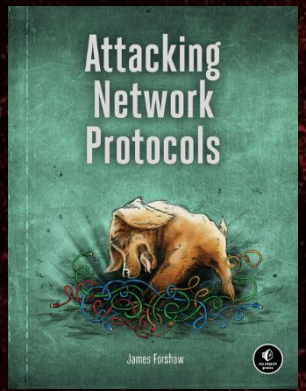
- Tutorials on how to use CANAPE with those scripts step-by-step:

  https://github.com/koutto/canape-resources/wiki

  Writing in progress…

- Future book by James Forshaw, Dec 2017:

  Probably a good read !



Snoopcon 2017


Snoopcon 2017
Questions ?