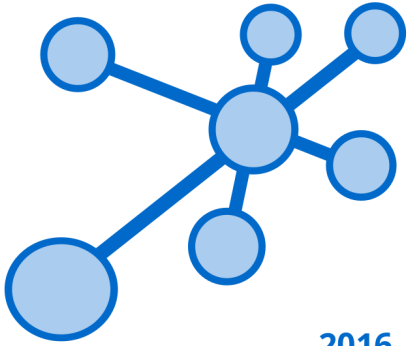
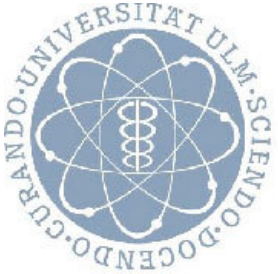


Introduction into Automatic Protocol Reverse Engineering

Sergej ‘winnie’ Schmidt

sergej.schmidt@uni-ulm.de
@violentinternet



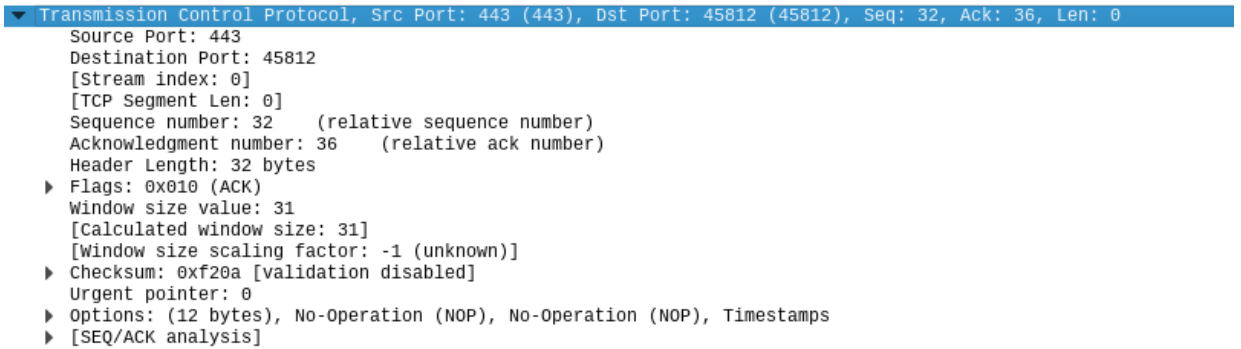
Motivation – Why PRE?

- Reversing tools (network stack)
- Vulnerability Hunting (Fuzzing)
- Botnet Analysis (Replaying)

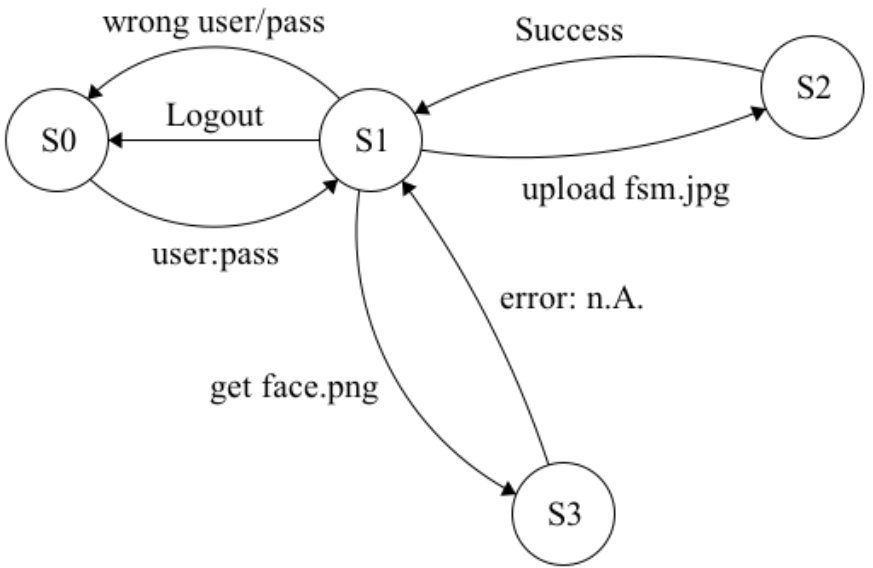
Protocol Definition

- Vocabulary (Messages/Packets)
  - We'll focus on that today
- Grammar (Message Order/State)

Vocabulary


Transmission Control Protocol, Src Port: 443 (443), Dst Port: 45812 (45812), Seq: 32, Ack: 36, Len: 0
Source Port: 443
Destination Port: 45812
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 32 (relative sequence number)
Acknowledgment number: 36 (relative ack number)
Header Length: 32 bytes
Flags: 0x010 (ACK)
Window size value: 31
[Calculated window size: 31]
[Window size scaling factor: -1 (unknown)]
Checksum: 0xf20a [validation disabled]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[SEQ/ACK analysis]

Grammar

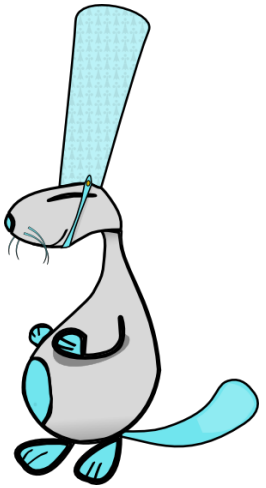

wrong user/pass
Success
S2
Logout
S0
S1
upload fsm.jpg
user:pass
error: n.A.
get face.png
S3

Approaches

- Network Trace Analysis (Netzob)
- Dynamic Binary Analysis (Polygot Paper)

Netzob

- Netzob[0]: Python2 lib
- Sequence Alignment with Needleman-Wunsch
- Clustering with UPGMA



Sequence Alignment

7	1	3	0	2	4
7	1	5	2	6	

7	1	3	0	2	4
7	1	5		2	6

Sequence Alignment

7	1	3	0	2	4
7	1	5	2	6	

+2	+2	-2	-2	-2	-1

7	1	3	0	2	4
7	1	5		2	6

+2	+2	-2	-1	+2	-2

Sequence Alignment

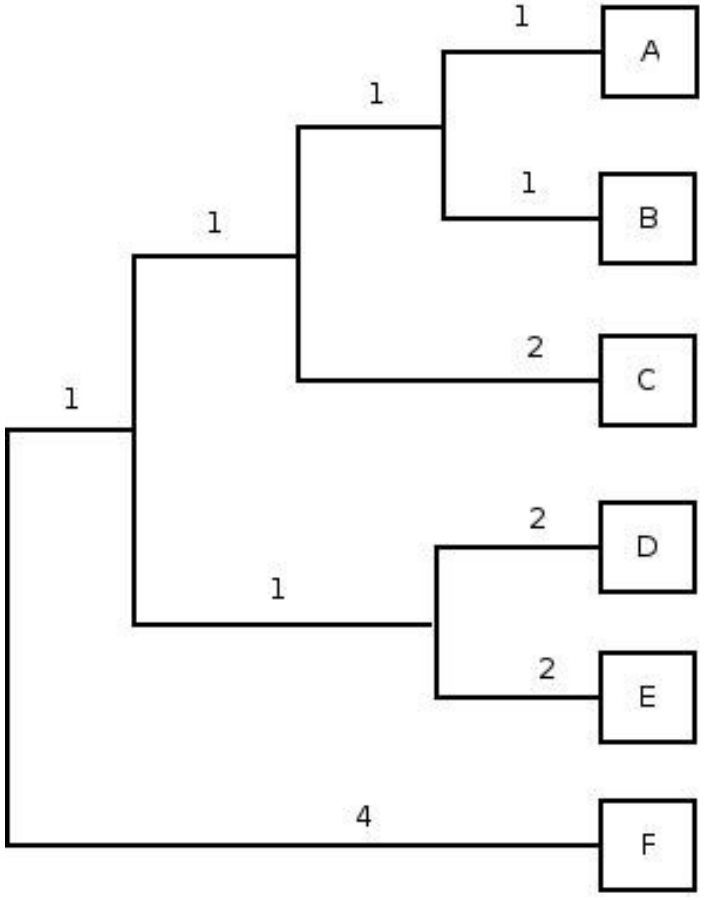
7	1	3	0	2	4
7	1	5	2	6	
+2	+2	-2	-2	-2	-1

Similarity
Score = -3

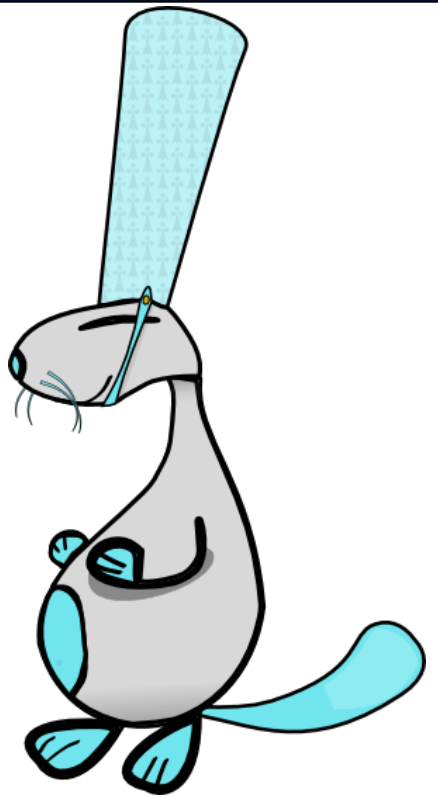
7	1	3	0	2	4
7	1	5		2	6
+2	+2	-2	-1	+2	-2

Similarity
Score = 1

UPGMA Clustering



Netzob - Demo



Netzob – Get Started

- https://netzob.readthedocs.org
  - Version “latest”, Tutorials are good.
- netzob.org not fully up2date
- Get Code:
  - https://dev.netzob.org/git/netzob.git
  - Synced on https://github.com/netzob/netzob
- Mailinglist used but not active, use IRC #netzob on freenode

Netzob – Project State

- Last stable version 0.4.1
  - Lots of features
  - Very nice GUI
  - Very confusing code
  - Annoyingly buggy

Netzob – Project State(2)

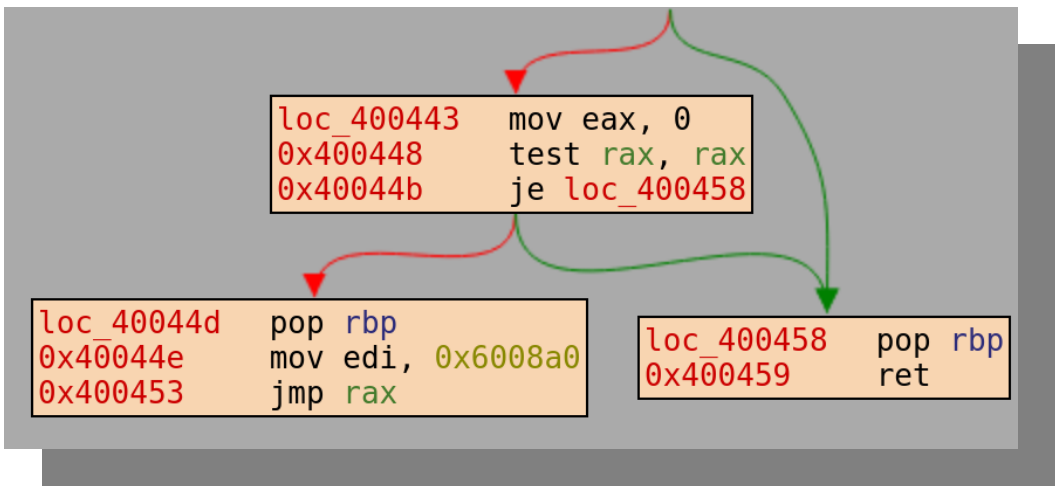
- Current v1.0rc1
  - Python Lib
  - Still some rough edges, flexible though
  - Does not Scale (Backtracing)
  - Very transparent and well documented code
  - Proper testing framework

Project Future

- Some Bugs
- Porting to Python v3
- Backport of old features:
  - Importers(XML, cleatext files, IPC)
  - Exporters(Peach, Wireshark, Sulley, Scapy)
- Stable Release Soon!?

Polygot - Dynamic Binary Analysis

- Dynamic Execution Trace (e.g. with qemu or qira[2])
- Memory Tainting
- Offline Analysis of Trace



Memory Tainting

- Mark and track input / output values through execution
- Mark values
  - Know the in-/out-coming values
  - Disassemble binary, find first input buffer
  - Look for typical socket syscalls, grab the buffer

```
connect(sockfd, &serv_addr, sizeof(serv_addr))
write(sockfd, &buffer, len_of_buffer);
read(sockfd, &buffer, len_of_buffer);
```

Analyzing fields

- Direction field (e.g. length)
  - Incremented after access to another field
- Field separators (e.g. \r\n)
  - Extract compared/accessed non-tainted values
  - If access occurs multiple times => match

Analyzing fields

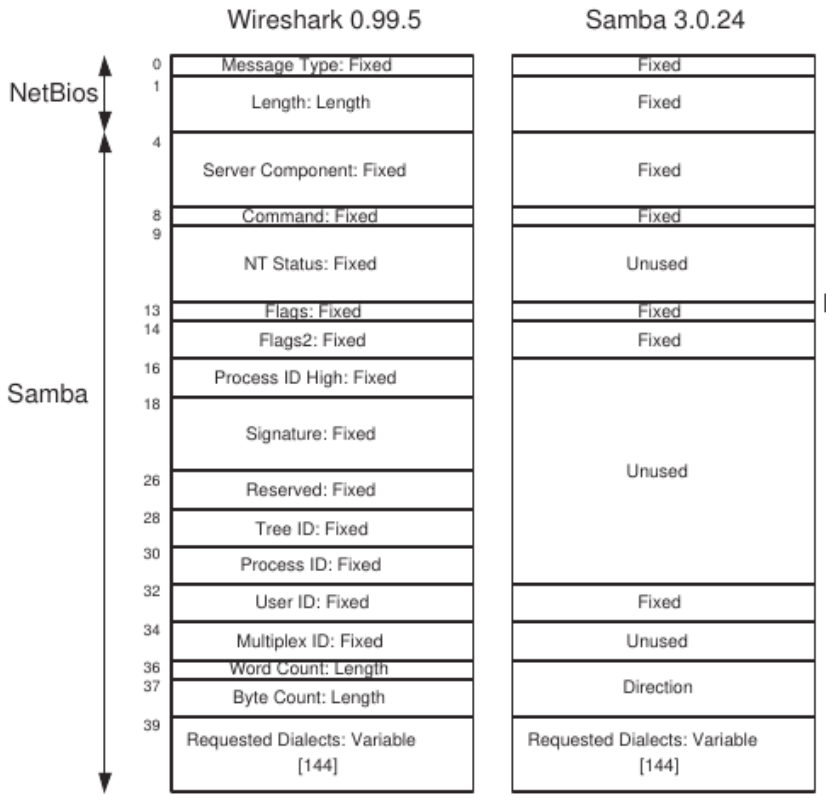
- Keywords
  - Fixed values like “Cookie” in HTTP
  - Comparisons of tainted and non-tainted memory
  - Only comparisons which yield true/false

Analyzing fields

- Fixed length field
  - Access to tainted memory area

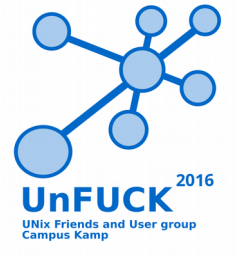
    movb EAX, [ESI] # ESI points to tainted
- Variable length fields
  - Analyze targets of direction fields

Results[1]


Wireshark 0.99.5
Samba 3.0.24
NetBios
Samba
0
Message Type: Fixed
Fixed
1
Length: Length
Fixed
4
Server Component: Fixed
Fixed
8
Command: Fixed
Fixed
9
NT Status: Fixed
Unused
13
Flags: Fixed
Fixed
14
Flags2: Fixed
Fixed
16
Process ID High: Fixed
18
Signature: Fixed
Unused
26
Reserved: Fixed
28
Tree ID: Fixed
30
Process ID: Fixed
32
User ID: Fixed
Fixed
34
Multiplex ID: Fixed
Unused
36
Word Count: Length
37
Byte Count: Length
Direction
39
Requested Dialects: Variable [144]
Requested Dialects: Variable [144]

Conclusion

- Got pcap or binary?
- Not many implementations (Netzob and Reverx)
- Netzob highly depends on network trace
  - Does not scale :-/
- Dynamic approach in general accurate



return “\0” || answers()

References/Links

- [0] https://netzob.org
  https://netzob.readthedocs.org

[1] Polygot: http://bitblaze.cs.berkeley.edu/protocol.html

[2] https://github.com/BinaryAnalysisPlatform/qira/