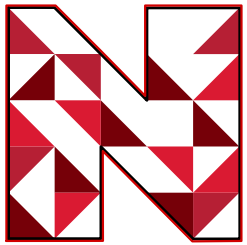
PCAP Feature Engineering for Machine Learning

whoami

Heather Lawrence

- Cyber Data Scientist @ NARI
- @infosecanon



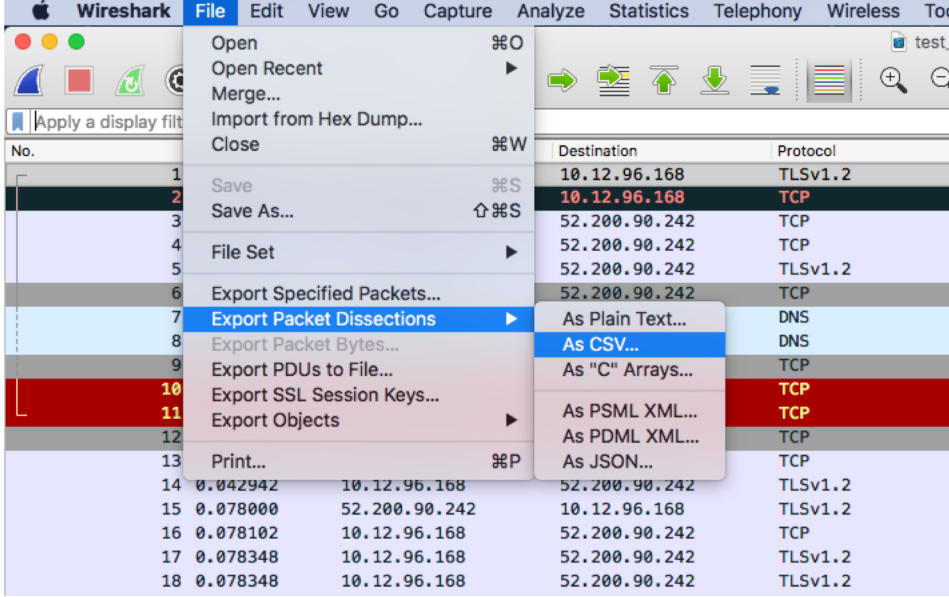
NEBRASKA APPLIED RESEARCH INSTITUTE

at the University of Nebraska

Outline

- Common pcap features
- Data normalization
- Packet inspection
- Timeline for TLS
- Why TLS v1.3
- Metadata
- Research using metadata
- Takeaways

Pcap -> csv



1054559	1554.028936	192.168.1.165	192.168.1.11	UDP	1494	35772 > 49156 Len=1452
1054560	1554.029009	192.168.1.165	192.168.1.11	UDP	1494	35772 > 49156 Len=1452
1054561	1554.029134	192.168.1.165	192.168.1.11	UDP	1494	35772 > 49156 Len=1452
1054562	1554.029136	192.168.1.122	192.168.1.201	TCP	91	4911 > 1167 [PSH, ACK] Seq=1673923 Ack=2505969 Win=33580 Len=37
1054563	1554.029136	192.168.1.122	192.168.1.201	TCP	91	[TCP Retransmission] 4911 > 1167 [PSH, ACK] Seq=1673923 Ack=2505969 Win=33580 Len=37
1054564	1554.029209	192.168.1.165	192.168.1.11	UDP	1494	35772 > 49156 Len=1452
1054565	1554.029210	192.168.1.165	192.168.1.11	UDP	1494	35772 > 49156 Len=1452
1054566	1554.029221	192.168.1.201	192.168.1.122	TCP	54	1167 > 4911 [ACK] Seq=2505969 Ack=1673960 Win=63253 Len=0
1054567	1554.029334	192.168.1.201	192.168.1.122	TCP	60	[TCP Dup ACK 1054566#1] 1167 > 4911 [ACK] Seq=2505969 Ack=1673960 Win=63253 Len=0
1054568	1554.029335	192.168.1.165	192.168.1.11	UDP	1494	35772 > 49156 Len=1452
1054569	1554.029336	192.168.1.165	192.168.1.11	UDP	1494	35772 > 49156 Len=1452
1054570	1554.029459	192.168.1.165	192.168.1.11	UDP	1494	35772 > 49156 Len=1452
1054571	1554.029461	192.168.1.165	192.168.1.11	UDP	1494	35772 > 49156 Len=1452

Common features from pcaps

- Duration
- Source and Destination IPs
- Protocol
- Length
- Payload

In [6]:
```
# Read in data from CSV
import os
os.chdir('/Users/z/Python/Baseline Prediction')
df = pd.read_csv("atk.csv")
df.head()
```

Out[6]:

	No	Time	Source	Destination	Protocol	Length	Info
0	1	0.000000	192.168.1.165	192.168.1.11	UDP	694	35774 > 49158 Len=652
1	2	0.000017	192.168.1.165	192.168.1.11	UDP	694	35774 > 49158 Len=652
2	3	0.029337	192.168.1.165	192.168.1.11	UDP	694	35774 > 49158 Len=652
3	4	0.029410	192.168.1.165	192.168.1.11	UDP	694	35774 > 49158 Len=652
4	5	0.052768	192.168.1.201	192.168.1.128	TCP	91	1172 > 4911 [PSH, ACK] Seq=1 Ack=1 Win=63290...

Data normalization

- Duration of conversation -> Numerical
- Source and Destination IPs -> Categorical
- Protocol -> Group by importance (binning)
- Length -> Numerical
- Payload -> Depends

Data normalization/feature eng

- Numerical -> average, min, median, max, entropy
- Categorical -> One-Hot Encoding
- Binning -> place into categories
    - Ex: all web traffic might include ports 80 and 443
- “Depends” -> What are you looking for?
    - Strings? First couple bytes for exe?

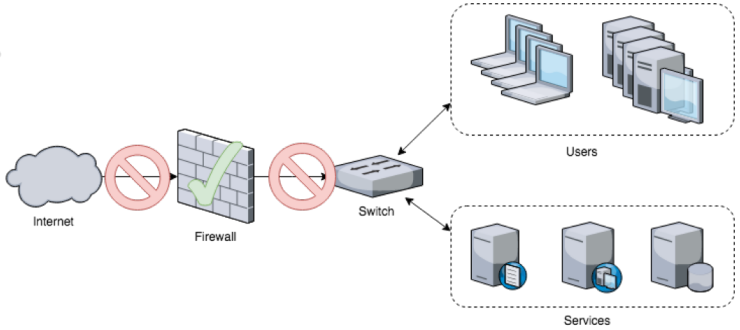
Data normalization/feature eng

- One-Hot

	Source	**Destination**
1	192.168.1.3	192.168.1.10
2	192.168.1.4	192.168.1.10

	192.168.1.3_s	**192.168.1.4_s**	**192.168.1.10_s**
1	0	0	1
2	0	1	0

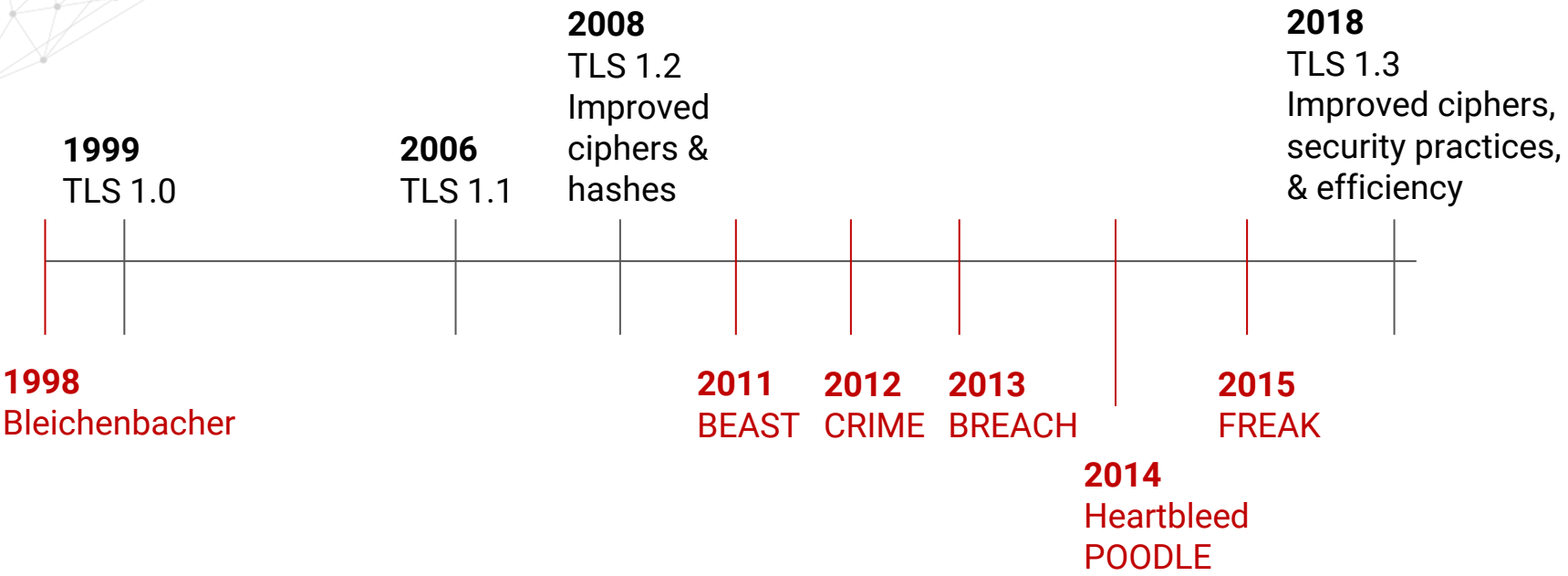
Packet Inspection




```
New TCP connection #35: 192.168.1.9(56006) <-> 192.168.1.7(445)
0.0002 (0.0002)  C>S
------------------------------------------------------------------
00 00 00 9b ff 53 4d 42 72 00 00 00 00 18 53 c8    .....SMBr.....S.
00 00 00 00 00 00 00 00 00 00 00 00 ff ff ff fe    ................
00 00 00 00 00 78 00 02 50 43 20 4e 45 54 57 4f    .....x..PC NETWO
52 4b 20 50 52 4f 47 52 41 4d 20 31 2e 30 00 02    RK PROGRAM 1.0..
4c 41 4e 4d 41 4e 31 2e 30 00 02 57 69 6e 64 6f    LANMAN1.0..Windo
77 73 20 66 6f 72 20 57 6f 72 6b 67 72 6f 75 70    ws for Workgroup
73 20 33 2e 31 61 00 02 4c 4d 31 2e 32 58 30 30    s 3.1a..LM1.2X00
32 00 02 4c 41 4e 4d 41 4e 32 2e 31 00 02 4e 54    2..LANMAN2.1..NT
20 4c 4d 20 30 2e 31 32 00 02 53 4d 42 20 32 2e     LM 0.12..SMB 2.
30 30 32 00 02 53 4d 42 20 32 2e 3f 3f 3f 00       002..SMB 2.???.
------------------------------------------------------------------
```

- Currently used by many NGFWs, IPS, sandboxes, network forensics, and network-based security analytics products. [8]

Timeline for TLS



Why TLS v 1.3?

- Adjustment to remove MiTM capabilities
  - Finance concerned over packet inspection [1][5]
- More of the negotiation handshake is encrypted [2]
- TLS 1.2 can be deployed securely
  - Several high profile vulnerabilities
  - Outdated algorithms
- TLS 1.3 removes the options and only supports protocols with no known vulnerabilities (PFS)
- Speed increases

If encryption is intended to protect end user communication, and breaking that encryption is considered a bad practice, can we detect malicious behavior without breaking the encryption?

Metadata

- While the `certificate` message is encrypted in TLS v1.3, side channel methods are still available
  - How frequently were they communicating?
  - How long was the conversation?
  - Size and distribution of bytes

Metadata

- Features can be taken from unencrypted `ClientHello` message [10]
  - Supported cipher suites
  - List of public keys accepted
  - Protocol versions supported

https://www.jgspiers.com/netscaler-client-certificate-ssl-handshake-failure-sha1-tls-1-2/

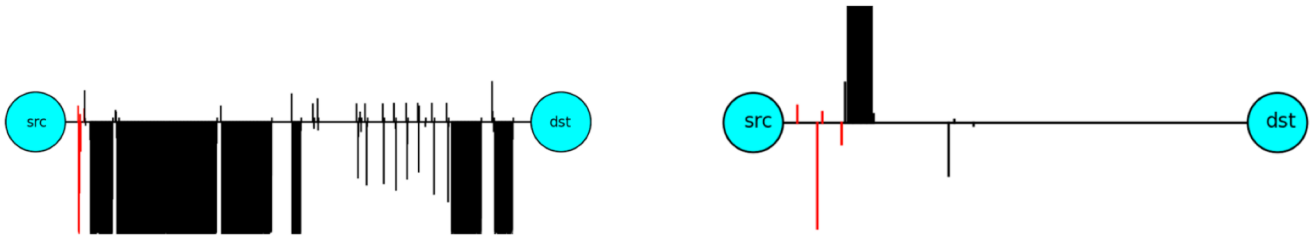
BotFinder: Finding Bots in Network Traffic Without DPI, Tegeler et al [7]

- Used ML to identify key features in C&C
- Trained models in a controlled environment to learn bot family behavior (timing patterns)
- Clustered 5 features:
  - Average time, avg duration, avg source bytes, avg destination bytes, and an identifiable frequency of communication (Fast Fourier)
- Tested on ISP network with billions of netflows

ML for Encrypted Malware Traffic...

B. Anderson, D. McGrew, 2017 [6]

- Analyzed accuracy of several algorithms
- Used min, mean, max, and standard deviation of:
  - Client -> server packet lengths
  - Server -> client packet lengths
  - Inter-arrival times
- Protocol, duration of network connection, and the number of packets and bytes sent/received
- Dataset contained millions of TLS encrypted sessions over 12 mo



(a) Google Search

(b) Bestafera

Figure 2: The TLS packet lengths and inter-arrival times for a typical Google search and malicious data exfiltration from `bestafera`. Upward and downward lines represent the sizes of packets being sent from client → server and server → client, respectively. The x-axis represents time.

B. Anderson, D. McGrew, 2017 [6]

Takeaways

- TLS v1.3 makes it harder to decrypt due to perfect forward secrecy (ECDHE)
- Losing DPI is not the end of the world
  - Just hasn’t been widely adopted yet
- Future network security technologies will likely focus on metadata extraction and evaluation over payload inspection and pattern matching

References

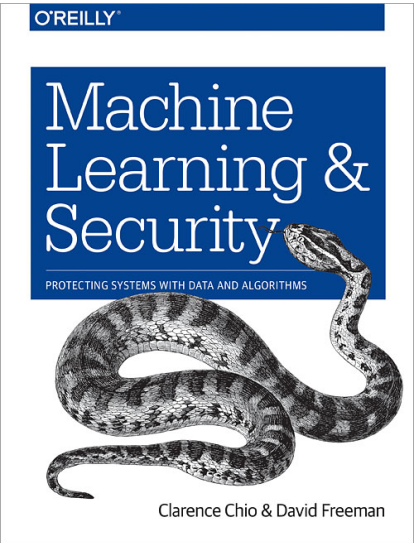
- [1] https://www.theregister.co.uk/2018/08/13/tls_13_approved/
- [2] https://www.ietf.org/blog/tls13/
- [3] https://scotthelme.co.uk/alexa-top-1-million-analysis-february-2019/
- [4] https://www.bleepingcomputer.com/news/security/ietf-approves-tls-13-as-internet-standard/
- [5] https://mailarchive.ietf.org/arch/msg/tls/CzjJB1g0uFypY8UDdr6P9SCQBqA
- [6] Anderson, B. & McGrew, D. (2017). *Machine Learning for Encrypted Malware Traffic Classification: Accounting for Noisy Labels and Non-Stationarity.* Paper presented at KDD'17 Halifax, NS Canada. DOI: 10.1145/3097983.3098163
- [7] Tegeler, F., Fu, X., Vigna, G., Kruegel, C. (2012). *BotFinder: Finding Bot in Network Traffic Without Deep Packet Inspection.* Paper presented at Co-NEXT'12 in Nice, France. DOI: 10.1145/2413176.2413217
- [8] https://www.darkreading.com/endpoint/tls-13-a-good-news-bad-news-scenario/a/d-id/1334180
- [9] https://assets.extrahop.com/whitepapers/EMA-ExtraHop-TLS13-2019-RR-SUMMARY.pdf

References

- [10] https://tls13.ulfheim.net/
- [11] https://blog.cloudflare.com/rfc-8446-aka-tls-1-3/
- [12] https://www.jgspiers.com/netscaler-client-certificate-ssl-handshake-failure-sha1-tls-1-2/

Resources

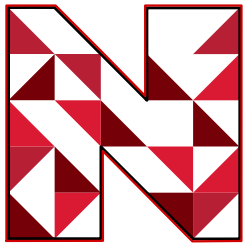
- Data Science Courses
  - Machine Learning & Security
  - https://www.coursera.org/learn/introduction-tensorflow
  - https://www.coursera.org/learn/machine-learning
  - http://course18.fast.ai/ml
- CDS People to follow on Twitter:
  - @KirkDBorne (Principal DS @BoozAllen)
  - @lorrietweet (Researcher, Director of CyLab CMU)
  - @BecomingDataSci (Good beginner content)
  - @hardmaru (Research sci @ Google Tokyo)
  - @biggiobattista (Adversarial ML)



Thanks!

Heather Lawrence

- Cyber Data Scientist @ NARI
- @infosecanon



NEBRASKA APPLIED RESEARCH INSTITUTE

at the University of Nebraska