

Preventing botnets



- Traditional
 - ▣ code analysis and finding malware fingerprints

- Code/binary analysis is mostly **manual** and increasingly **harder**
 - ▣ Code obfuscation
 - ▣ Encryption
 - ▣ Self-modifying

- Behavior-based analysis is much harder to thwart
 - ▣ Bots need to **communicate!**

Preventing botnets



- Traditional
 - code analysis and finding malware fingerprints

- Code/binary analysis is mostly **manual** and increasingly **harder**
 - Code obfuscation
 - Encryption
 -

- B h
 - bots need to **communicate**:

Lots of data
available...

Binary code (Zeus)

```
4D 5A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 D8 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 50 45 00 00 4C 01 03 00 75
39 B2 52 00 00 00 00 00 00 00 00 00 00 E0 00 02 01 0B 01 0A 00 00 08 02 00 00 3A
00 00 00 00 00 00 48 30 01 00 00 10 00 00 00 20 02 00 00 00 40 00 00 10 00
00 00 02 00 00 05 00 01 00 01 00 00 00 05 00 01 00 00 00 00 00 70 02 00
00 04 00 00 00 00 00 00 00 02 00 00 81 00 00 10 00 00 10 00 00 00 10 00 00
10 00 00 00 00 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00 A4 F7 01 00 18 01
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 50 02 00 AC 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 10 00 00 A0 05 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 2E 74 65 78 74 00 00 00 84 06 02
00 00 10 00 00 00 08 02 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 00 00 60 2E 64 61 74 61 00 00 00 50 20 00 00 00 20 02 00 00 04 00 00 00
0C 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 C0 2E 72 65 6C 6F 63
00 00 7C 16 00 00 00 50 02 00 00 18 00 00 00 10 02 00 00 00 00 00 00 00 00
00 00 00 00 00 40 00 00 42 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
```

Network traffic (HLUX2)

The image shows a screenshot of the Wireshark network traffic analysis tool. The main window displays a list of captured packets. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, Internals, and Help. The toolbar contains various icons for navigation and analysis. The filter bar is set to 'Expression...'. The packet list table has columns for No., Time, Source, Destination, Protocol, Length, and Info. The selected packet (No. 1) is expanded to show its details: Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
148	67.056100	78.97.23.219	81.167.148.237	TCP	60	80-1122 [FIN, ACK] Seq=1 Ack=2 Win=65535 Len=0
149	67.059054	81.167.148.237	78.97.23.219	TCP	60	1122-80 [ACK] Seq=2 Ack=2 Win=64240 Len=0
150	69.639592	81.167.148.237	61.117.196.249	TCP	62	[TCP Retransmission] 1124-80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
151	75.646825	81.167.148.237	61.117.196.249	TCP	62	[TCP Retransmission] 1124-80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
152	76.034688	61.117.196.249	81.167.148.237	TCP	62	80-1124 [SYN, ACK] Seq=0 Ack=1 Win=26280 Len=0 MSS=1460 SACK_PERM=1
153	76.040994	81.167.148.237	61.117.196.249	TCP	60	1124-80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
154	76.539499	81.167.148.237	61.117.196.249	TCP	60	1124-80 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
155	76.599564	81.167.148.237	93.118.210.100	TCP	62	1125-80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
156	76.680915	93.118.210.100	81.167.148.237	TCP	62	80-1125 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 SACK_PERM=1
157	76.684395	81.167.148.237	93.118.210.100	TCP	60	1125-80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
158	77.030573	61.117.196.249	81.167.148.237	TCP	60	80-1124 [ACK] Seq=1 Ack=2 Win=26280 Len=0
159	77.178882	81.167.148.237	93.118.210.100	TCP	60	1125-80 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
160	77.239849	81.167.148.237	82.115.77.39	TCP	62	1126-80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
161	77.257201	93.118.210.100	81.167.148.237	TCP	60	80-1125 [ACK] Seq=1 Ack=2 Win=64240 Len=0
162	77.257241	93.118.210.100	81.167.148.237	TCP	60	80-1125 [FIN, ACK] Seq=1 Ack=2 Win=64240 Len=0
163	77.260790	81.167.148.237	93.118.210.100	TCP	60	1125-80 [ACK] Seq=2 Ack=2 Win=64240 Len=0
164	77.290345	82.115.77.39	81.167.148.237	TCP	62	80-1126 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM=1
165	77.293390	81.167.148.237	82.115.77.39	TCP	60	1126-80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
166	77.790466	81.167.148.237	82.115.77.39	TCP	60	1126-80 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
167	77.840880	82.115.77.39	81.167.148.237	TCP	60	80-1126 [ACK] Seq=1 Ack=2 Win=65535 Len=0
168	77.851215	81.167.148.237	130.204.162.56	TCP	62	1127-80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
169	77.928675	130.204.162.56	81.167.148.237	TCP	62	80-1127 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM=1
170	77.933320	81.167.148.237	130.204.162.56	TCP	60	1127-80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
171	78.430946	81.167.148.237	130.204.162.56	TCP	60	1127-80 [FIN, ACK] Seq=1 Ack=1 Win=64240 Len=0
172	78.492819	81.167.148.237	158.181.141.49	TCP	62	1128-80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
173	78.508706	130.204.162.56	81.167.148.237	TCP	60	80-1127 [ACK] Seq=1 Ack=2 Win=65535 Len=0
174	79.028212	61.117.196.249	81.167.148.237	TCP	60	80-1124 [FIN, ACK] Seq=1 Ack=2 Win=26280 Len=0

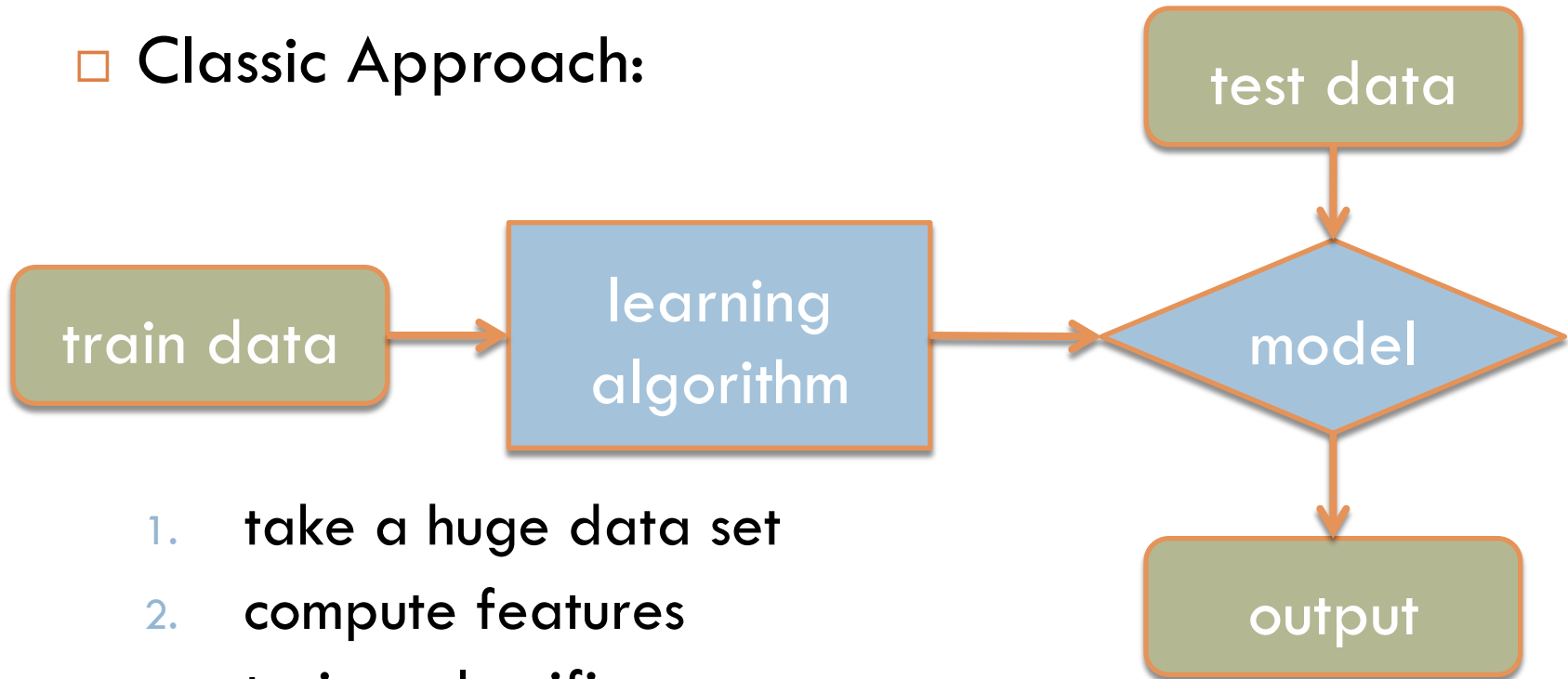
Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
Ethernet II, Src: CadmusCo_94:a7:89 (08:00:27:94:a7:89), Dst: Cisco_f1:ad:80 (00:0a:41:f1:ad:80)
Internet Protocol Version 4, Src: 81.167.148.237 (81.167.148.237), Dst: 173.29.103.45 (173.29.103.45)
Transmission Control Protocol, Src Port: 1100 (1100), Dst Port: 80 (80), Seq: 0, Len: 0

```
0000 00 0a 41 f1 ad 80 08 00 27 94 a7 89 08 00 45 00  ..A.....E.  
0010 00 30 01 ff 40 00 80 06 fd e9 51 a7 94 ed ad 1d  .0..@.....Q.....  
0020 67 2d 04 4c 00 50 c7 21 03 3b 00 00 00 00 70 02  g-.L.P!.....p.  
0030 fa f0 be 56 00 00 02 04 05 b4 01 01 04 02      ...V.....
```

File: "/Users/sicco/USB/hlux... | Packets: 38511 · Displayed: 38511 (100.0%) · Load time: 0:00.385 | Profile: Default

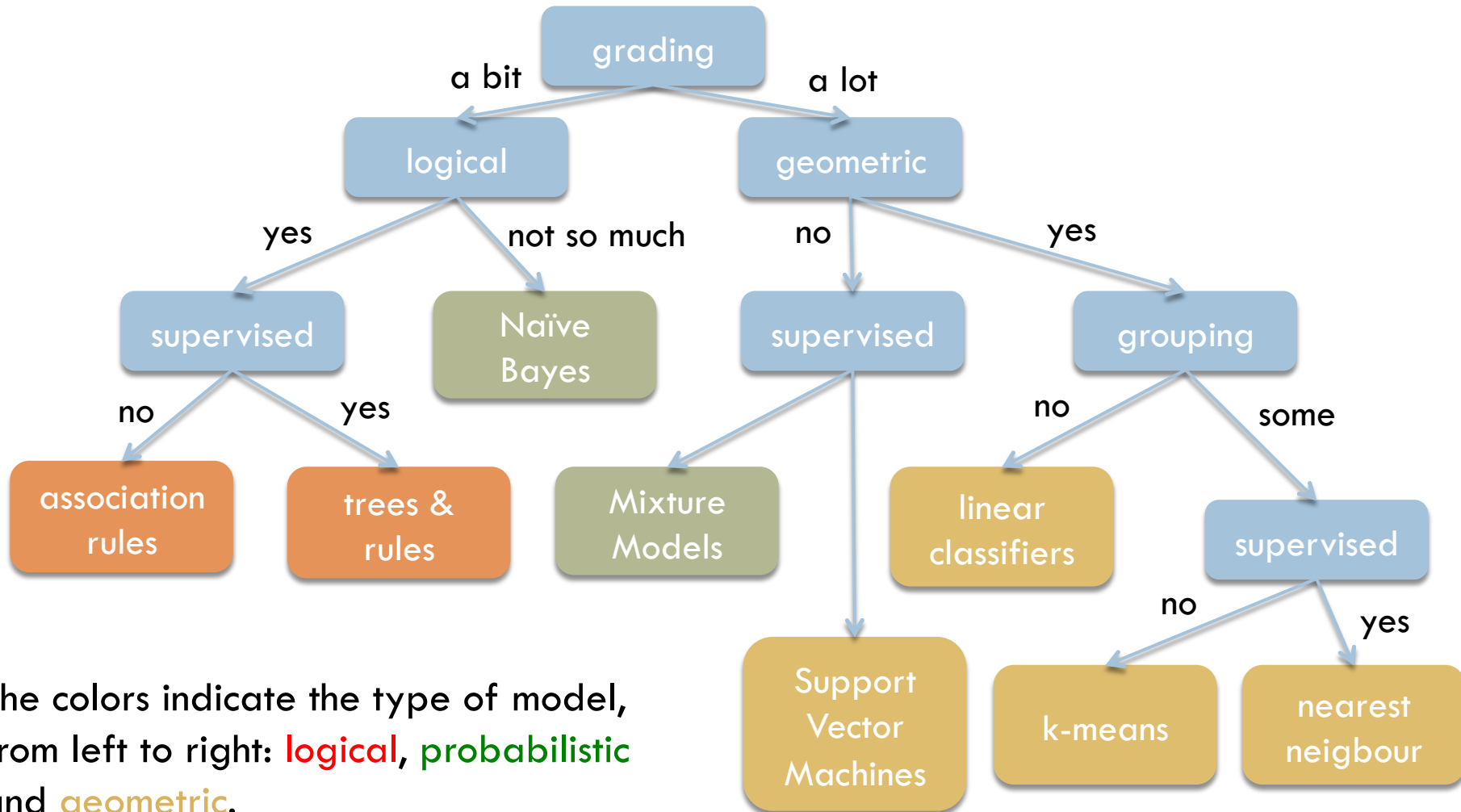
Machine learning

□ Classic Approach:



1. take a huge data set
2. compute features
3. train a classifier
4. deploy the classifier on test

ML Taxonomy – many approaches



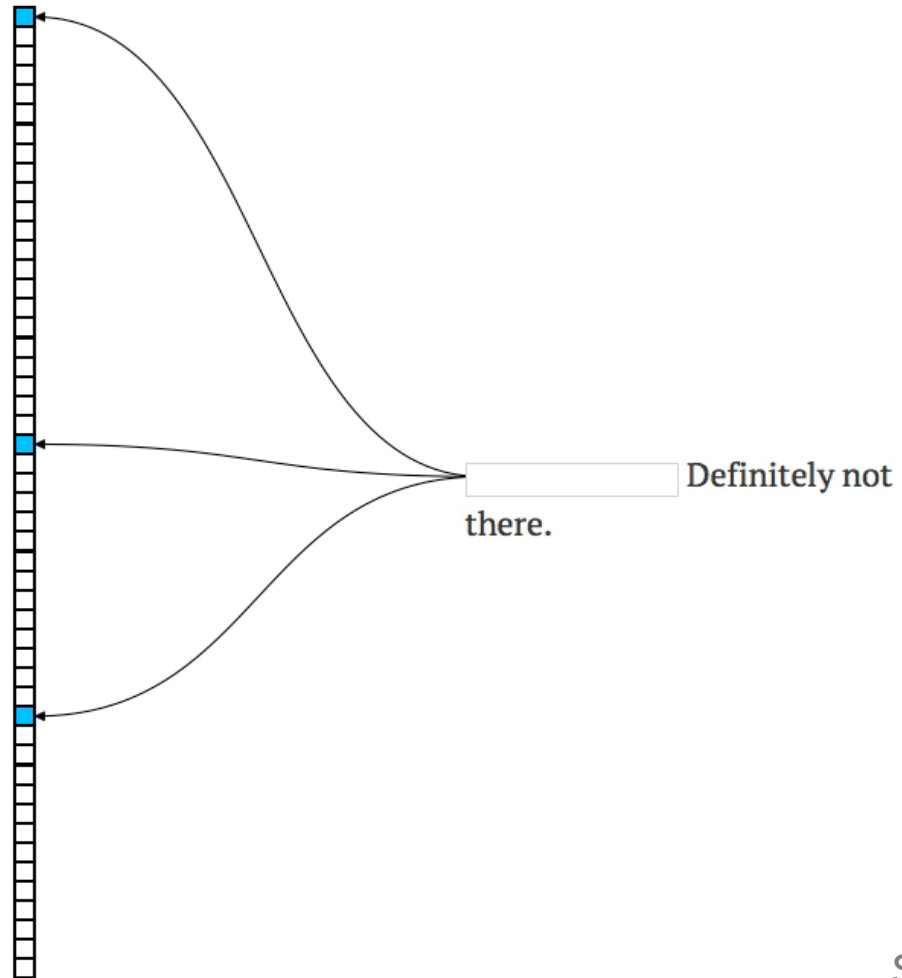
Classic ML fails in cyber security

- Large majority ($> 99\%$) of cases are benign!
 - ▣ adapt data/models, otherwise no positives
- Data is massive and keeps coming in!
 - ▣ need to count quickly, reduce false positives
- There is an opponent, they learn too!
 - ▣ avoid using generic fingerprints/simple rules
- Privacy makes data inaccessible...

Bloom filter: dealing with massive data

8

Key:



Course

- Read scientific papers, use techniques on real data:
 - Credit fraud data from Adyen
 - Botnet traffic from HLUX2
 - NetFlow traces, perhaps from EEMCS...
 - ...
- Learn by doing in labs, an exam on content