Security of Networks, Services, and Systems

Intrusion Detection Systems

Ricardo Morla

FEUP – SSR/M.EEC, SR/M.EIC

Types of intrusion detection

- HIDS vs. NIDS
- IDS vs. IPS
- Knowledge-based vs. behavior-based

Where are intrusions being detected?

- In a host
  - Host IDS
  - Host-based systems detect intrusions by looking at data from the operating system (memory, file-system, ...) and applications

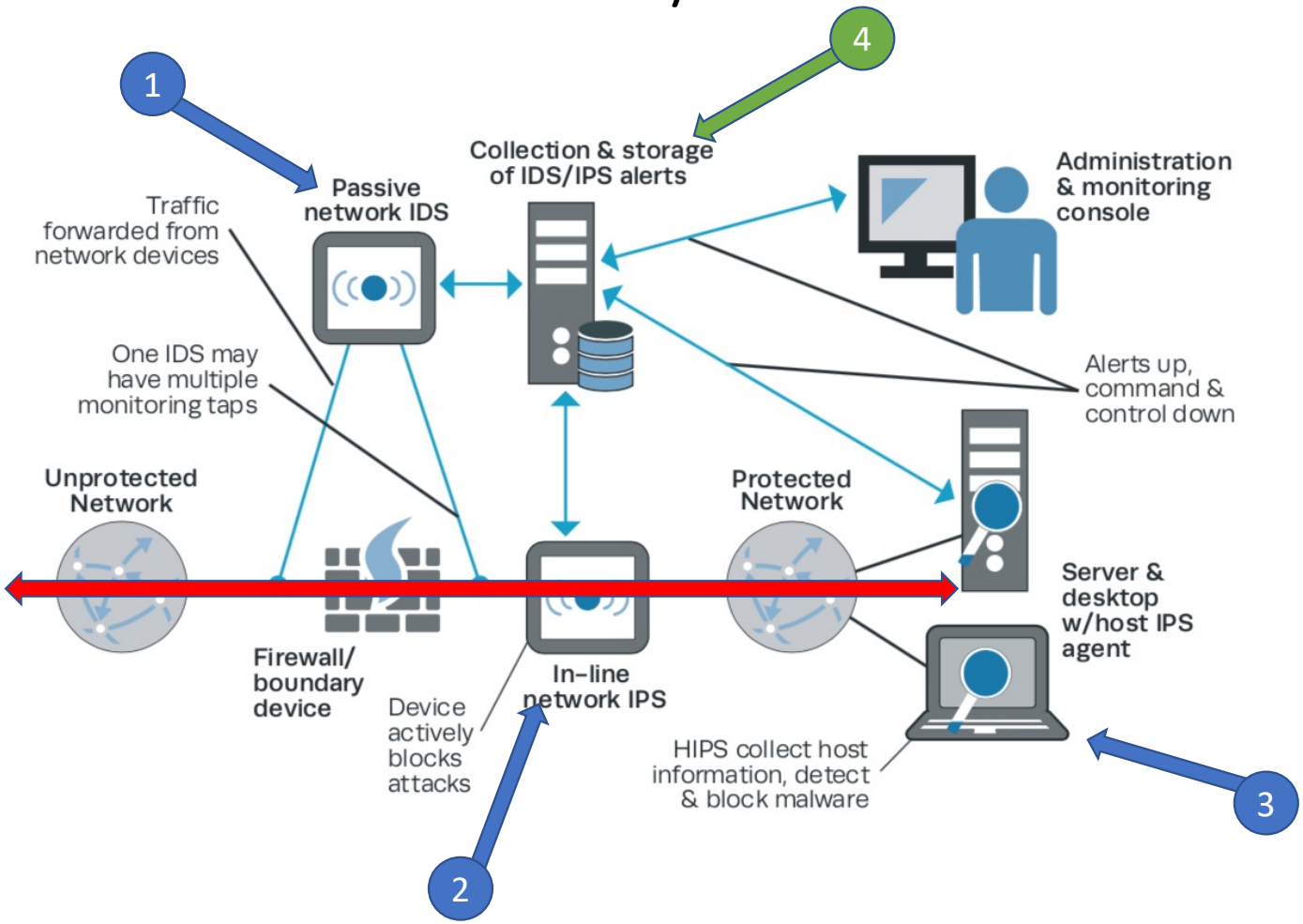
- In the network or at some network interface
  - NIDS, Network IDS
  - Network-based systems detect intrusions by looking at network traffic

What do you do when you detect?

- Nothing – not immediately at least
  - It's an intrusion **detection** system
  - Have larger time constant actions on the target system

- Drop packet, remove file, kill process, …
  - So that'll be an intrusion **prevention** system
  - Immediately respond to prevent intrusion

Location of Network IDS/IPS



How do you know what to detect?

- Knowledge-based
  - Encode pre-existing knowledge about intrusions into rules
  - How do you get gre-existing knowledge? Threat intel, etc
- Behavior-based
  - Learn from the data how to detect intrusions
  - Extract ‘knowledge’ from data

Knowledge-based IDS

- Protocol
  - `alert icmp any any -> any any (msg: "ICMP Packet found";)`
- Content
  - `alert tcp 192.168.1.0/24 any -> any any (content: "HTTP"; offset: 4; msg: "HTTP matched";)`
- SYN flood
  - `alert tcp any any -> $HOME_NET 80 (flags: S; msg:"Possible TCP DoS"; flow: stateless; detection_filter: track by_dst, count 70, seconds 10;)`

Knowledge-based IDS

- TLS fingerprinting (ja3, ja3s)
  - `alert tls any any -> any any (msg:"match JA3 hash"; ja3.hash; content:"e7eca2baf4458d095b7f45da28c16c34"; sid:100001;)`


```
{
    "id": 0,
    "desc": "ThunderBird (v38.0.1 OS X)",
    "record_tls_version": "0x0301",
    "tls_version": "0x0303",
    "ciphersuite_length": "0x0016",
    "ciphersuite": "0xC02B 0xC02F 0xC00A 0xC009 0xC013 0xC014 0x0033 0x0039 0x002F 0x0035 0x000A",
    "compression_length": "1",
    "compression": "0x00",
    "extensions": "0x0000 0xFF01 0x000A 0x000B 0x0023 0x0005 0x000D 0x0015" ,
    "e_curves": "0x0017 0x0018 0x0019" ,
    "sig_alg": "0x0401 0x0501 0x0201 0x0403 0x0503 0x0203 0x0402 0x0202" ,
    "ec_point_fmt": "0x00"
}
```


Some knowledge-based Network IDS/IPS tools

- Suricata
  - https://suricata.io
- Snort
  - https://www.snort.org
- Zeek
  - https://github.com/zeek/zeek

Knowledge-based IPS vs. Firewall

- How different are IPS's from Firewalls?
- Firewalls also have rules
- Application-layer firewalls can also inspect packet payloads

- Maybe knowledge-based IPS's are not very different from application-layer firewalls
  - Maybe firewalls are most of the times based on rules, which is a significant difference to behavior-based IPS's

Behavior-based IDS

- Anomaly detection, often relying on learning from data
  - But could also be classification
- Problems
  - Defining normal behavior hard, training data
  - Normal behavior and traffic drifting
  - Excessive sensitivity and high false positive rate
  - Adversarial traffic can blind the IDS, detecting normal traffic as intrusion and vice-versa
  - Root-cause analysis to detect attack source and adequately stop intrusion

Learning from data

- Learn model for normal behavior
  - Unsupervised learning, no labels
  - Apply outlier detection techniques to find intrusions
- Learn model that distinguishes normal from intrusion
  - Classification problem
  - Need to label normal and intrusion data
  - More normal data than intrusion can cause problems in learning

Network data types

- Packet traces
- Counters
- Flow data
- Others

- In any case, need structured data set as input for learning problem

#1 Packet traces

- Motion-picture-like recording of everything that goes through the network
  - What, when, where, who (?), why (?)
- Raw data – powerful but hard to use
- Difficult to manage
  - Capture limitations (copy data at 1, 10, etc Gbps scale)
  - Storage limitations (Gbps * minutes, hours, days = ?)
- Difficult to use and process
  - Not in a table like format – would be easier
  - Can write processing rules to create tables – but only partial vision
  - AI and deep learning etc to process traces (raw or features)

#2 Traffic counters on links

- Routers keep track of how much traffic goes through each link
  - Packets, bytes
  - Periodically every n minutes
- Simple to use but limited in scope
  - Coarse metrics
  - Link failures, capacity DoS

#3 Flow measurements

- IP flows
  - Source, destination IP address and TCP/UDP ports (4 fields)
  - L3 header protocol (TCP, UDP, other)
  - Other info – ToS field, ??
- Keeps record of traffic for each flow
  - Packet, byte count on each direction
  - Duration, first/last packet timestamps, TCP flags
  - etc
- Tradeoff
  - simpler to use than packet traces
  - more information than counters
  - simply opening a web page can generate dozens of TCP flows

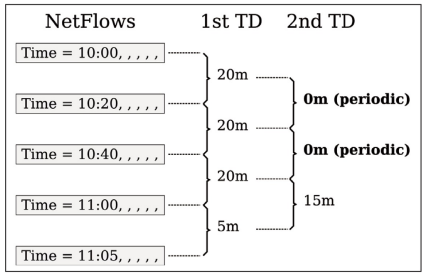
Other data

- Extract custom information from packet traces
- Tshark, the command line version of wireshark, other tools
- Examples:
  - sequence of TLS records per flow
  - flags for specific protocols
  - domain name queries
  - …

Stratosphere IPS

S. Garcia, *Modelling the Network Behaviour of Malware To Block Malicious Patterns. the Stratosphere Project: a Behavioural IPS" in Virus Bulletin*, pp. 1-8, September 2015. https://www.virusbulletin.com/uploads/pdf/conference/vb2015/Garcia-VB2015.pdf

- Feature extraction
  - 4-tuple object: Source IP, Destination IP+TCP port
  - Each letter represents a TCP connection belonging to the 4-tuple object



	Small Size			Medium Size			Big Size		
Duration	Short	Medium	Long	Short	Medium	Long	Short	Medium	Long
Not enough data	1	2	3	4	5	6	7	8	9
Strongly periodic	a	b	c	d	e	f	g	h	i
Weakly periodic	A	B	C	D	E	F	G	H	I
Not periodic	r	s	t	u	v	w	x	y	z

Botnet C2 host talking with C2 server

```
11aaaaaaaaaabrrctrraaaAaaaaaAaaaaaaaaaaaaaaaaaAAAa
aaaaaaaaaaaaaaaaaaAaAaaaaaaaaaaaaaaaaaaaaaAAaaaaa
aaaAAaaaaaaaaaaaaaaaaaaaaaaaaAAAAaaaaaaaaAAaaAAaaa
aaaaaaaaaaaaaaaaaaAAaaaaaaaaaAAaaaaaaaaaaaaaaaaaaa
aaaaaaaaaAAaAaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaAaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaAAaaaaaaaaaaaaaaaaaa
aaAAaaaaaaaaAaaAAAAAAAAaaaaaaaaaaaaaaaaaaaaaaaaaa
aa(...)
```

Host connecting to Google services

```
71r0rxrurradrrAAarraArudaurrDuradrrDdrraadurrgdruAxr
ruruDAxrDuarxraaduuadurrrruurdruurrG0rurarruurur0rur
0rruuurrrudrurraruadurrudruuaruxurarrrurauruururrruAdr
rruu0rrrurrraruaarrrurаauurrrarrrrrurrurrurrurrrrrrrx
uurrurrrrrurruuuurrrrrrruurrrarrauuarrarx0rruruaarruur
rrrauAaduuddruaaruuaaaruruarrDrrdruDrraaarrrDddxrrrd
axraDdurrrrururrrurrrrrrrrrurDrr(...)
```

Packet x-ray

Zhou, Z., Yao, L., Li, J., Hu, B., Wang, C., & Wang, Z. (2018). Classification of botnet families based on features self-learning under Network Traffic Censorship. *2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC)*, 1–7. https://doi.org/10.1109/SSIC.2018.8556792

- Classify botnet traffic
- First 400 IP payload bytes, TCP flows
- 20x20 pixels, 0-255
- CNN classifier
- Mostly good classification results

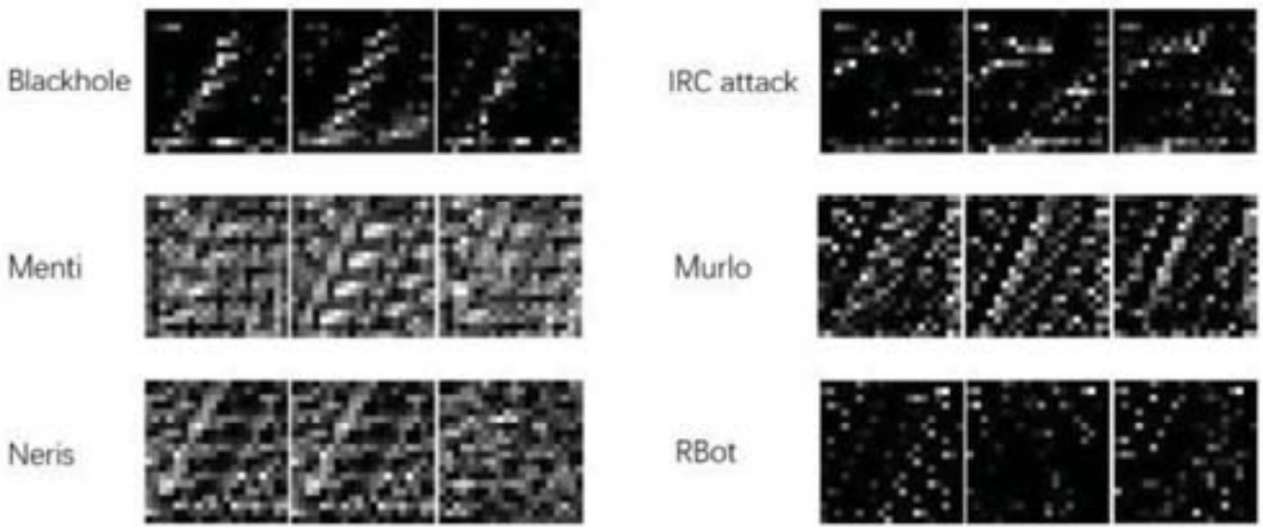
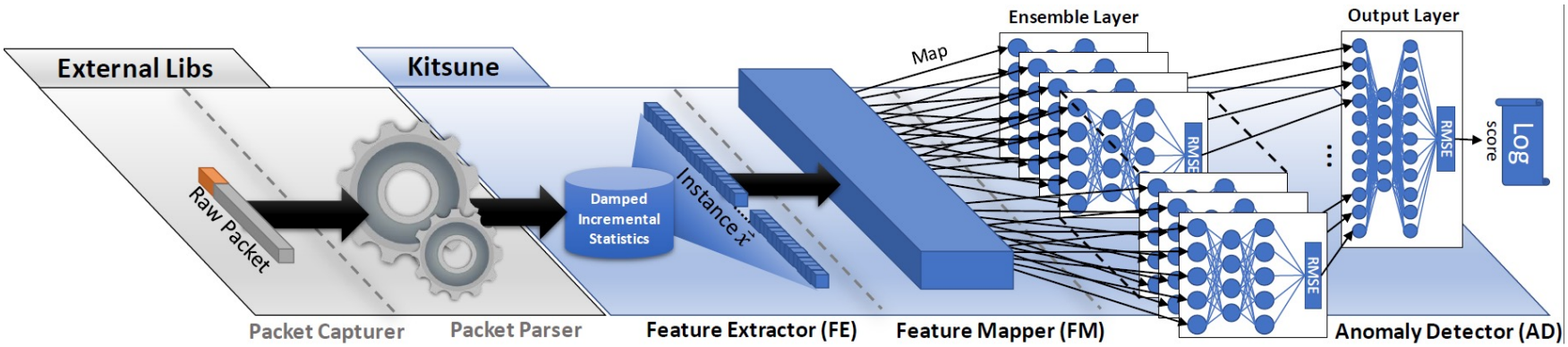


Fig. 8. Six samples of representation of the input data

Kitsune

https://github.com/ymirsky/Kitsune-py
https://arxiv.org/pdf/1802.09089.pdf

- General purpose online intrusion detection
  - unsupervised learning
  - Learns for 5 minutes, then starts detecting
  - Input: raw packet data



Many threats to detect in knowledge and behavior-based IDS

- Command and control
  - DGA, TLS, …
- Exfiltration
  - DNS, malformed packets, …
- Phishing
  - URLs
- DoS
  - Low intensity from botnets
- …

Security of Networks, Services, and Systems

Intrusion Detection Systems

Ricardo Morla

FEUP – SSR/M.EEC, SR/M.EIC