Lab 8: Firewalls & Intrusion Detection Systems

Fengwei Zhang

Firewall & IDS

- Firewall
  - A device or application that analyzes packet headers and enforces policy based on protocol type, source address, destination address, source port, and destination port. Packets that do not match policy are rejected
- Intrusion Detection System (IDS)
  - A device or application that analyzes whole packets, both header and payload, looking for known events. When a known event is detected, a log message is garneted detailing the event
- Intrusion Prevent System (IPS)
  - A device or application that analyzes whole packets, both header and payload, looking for known events. When a known event is detected, the packet is rejected
- Modern devices combines all of these functions in a single device/ application (Smart Firewall)

Types of IDS

- Host-based IDS (HIDS)
  - Installed locally on machines
  - Monitoring local user security
  - Monitoring program execution
  - Monitoring local system logs
- Network-based IDS (NIDS)
  - Sensors are installed on the network
  - Monitor network activity (deep packet inspection)

Types of Network-based IDS

- Signature-based IDS
  - Compares incoming packets with known signatures
  - E.g., Snort, Bro, Suricata
- Anomaly-based IDS
  - Leans the normal behavior of the system
  - Generates alerts on packets that are different form the normal behavior

Signature-based IDS

- Anti-virus tools
- Problems
  - “Zero-day” attacks
  - Polymorphic attacks

Anomaly-based IDS

- Anomaly-based IDS is capable of identifying "Zero-day" attacks

- Problems
  - High false positive rates
  - Labeled training data

IDS Evaluation Metrics

- True Positives (TP)
  - A genuine attack is detected
- True Negatives (TN)
  - Benign traffic identified as benign
- False Positives (FP)
  - Harmless behavior is misclassified as an attack
- False negatives (FN)
  - A genuine attack is not detected

- An intrusion detection system is:
  - Accurate: if it detects all genuine attacks
  - Precise: if it never reports legitimate behavior as an attack

IDS Evaluation Metrics

- The true positive rate is: TP / (TP + FN)
  - TP is the number of the true positives
  - FN is the number of the false negatives
  - TP + FN is the total number of positives

- The false positive rate: FP / (FP + TN)
  - FP is the number of the false positives
  - TN is the number of the true negatives
  - FP + TN is the total number of negatives

IDS Evaluation Metrics

- An undetected attack might lead to severe problems; frequent false alarms can lead to the system being disabled or ignored. A perfect IDS would be both accurate and precise
- Suppose that only 1% of traffic are actually attacks; the detection accuracy of your IDS is 90%; the false positive rate is 10%
- If you have an alarm, what is the chance that it is a false alarm?

IDS Evaluation Metrics

- Suppose that only 1% of traffic are actually attacks
  - 1000 events: 990 benign; 10 attacks
- The detection accuracy of your IDS is 90%
  - True positive rate: 90%
  - True positive number: 10*90%=9 true alarms
- The false positive rate is 10%
  - False positive rate: 10%
  - False positive number: 990*10%=99 false alarms

- P (attacks/alarms) = 9/(9+99) = 0.083333
- There is approximately 92% chance that a raised alarm is false

Snort

- Signature-based IDS
- Can be run as IPS or IDS
- First released in 1997 but still updated and maintained today
- Latest version Snort 2.9.8.2

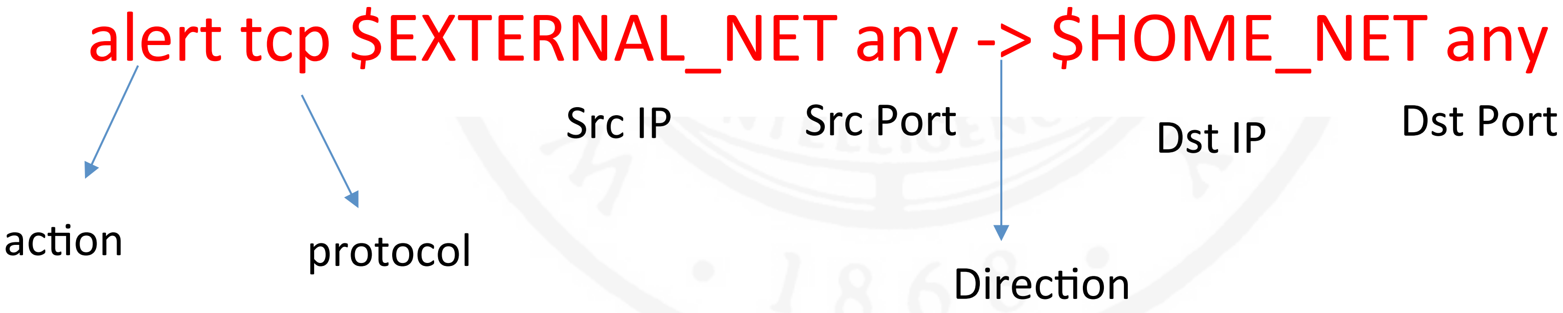
Snort Rules

```
alert tcp $EXTERNAL_NET any -> $HOME_NET
any (msg:"SCAN SYN FIN";flags:SF; reference:
arachnids,198; classtype:attempted-recon; sid:
624; rev:1;)
```

rule header (rule options)

Snort Rule Header

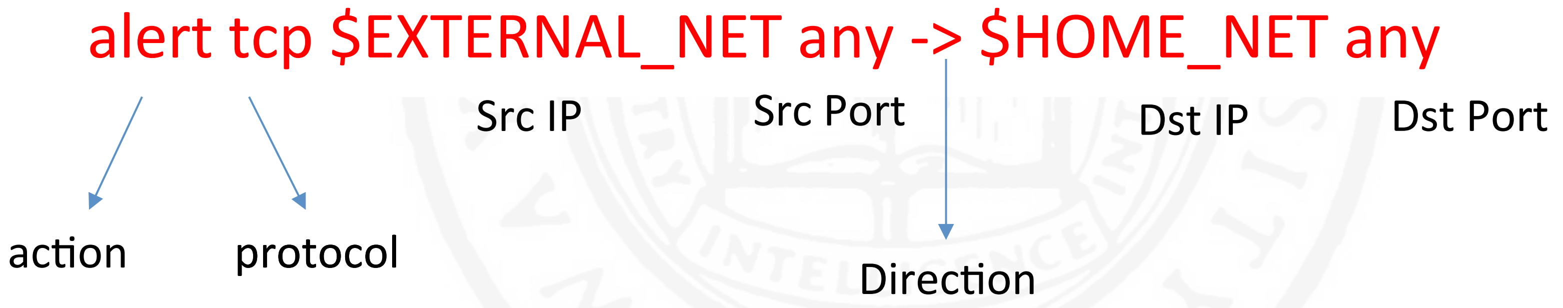
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN SYN FIN";flags:SF; reference: arachnids, 198; classtype:attempted-recon; sid:624; rev:1;)



alert tcp $EXTERNAL_NET any -> $HOME_NET any

Snort Rule Header Action

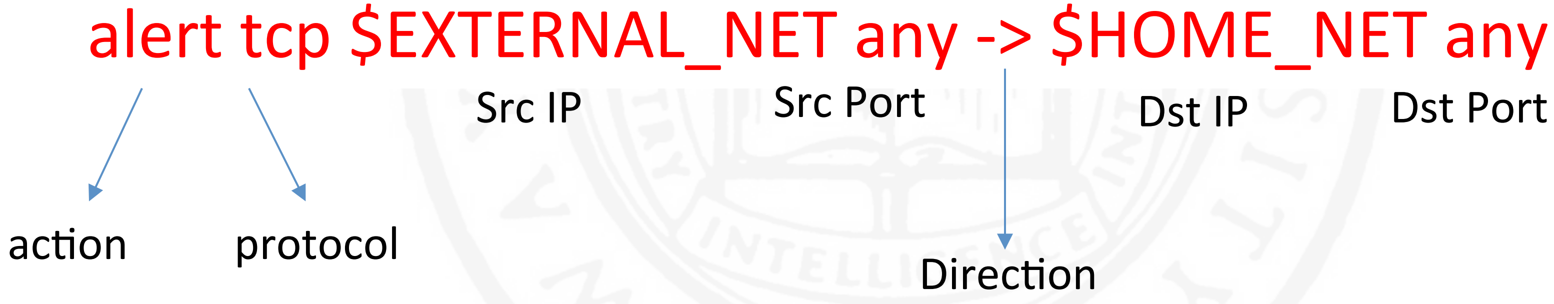
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN SYN FIN";flags:SF; reference: arachnids,198; classtype:attempted-recon; sid:624; rev:1;)


alert tcp $EXTERNAL_NET any -> $HOME_NET any

1. **alert**: Alerts and logs the packet when triggered.
2. **log**: Only logs the packet when triggered.
3. **pass** : Ignores or drops the packet or traffic matching.
4. **activate** : Alerts then activates a dynamic rule or rules.
5. **dynamic** : Ignores, until started by the activate rule, at which time, acts as a log rule.
6. **drop** : block and log the packet
7. **reject** : block the packet, log it, and then send a TCP reset if the protocol is TCP or an ICMP port unreachable message if the protocol is UDP.
8. **sdrop** : block the packet but do not log it.

Snort Rule Header Procotol

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any
(msg:"SCAN SYN FIN";flags:SF; reference: arachnids,
198; classtype:attempted-recon; sid:624; rev:1;)
```


alert tcp $EXTERNAL_NET any -> $HOME_NET any

Protocols: TCP, UDP, ICMP, and IP
Future may include: ARP, IGRP, GRE, OSPF, RIP, IPX, etc.

Snort Rule Header IP

```
          Src IP        Src Port Dst IP  Dst Port
alert tcp $EXTERNAL_NET any -> $HOME_NET any
alert tcp 192.168.1.0/24 any -> 192.168.1.0/24 1:1024
alert tcp ![192.168.1.0/24,10.1.1.0/24] any -> 192.168.1.44
```

- $EXTERNAL_NET is a config value set in snort.conf
- IP is specified also as dotted notation with CIDR masks. “any” is also valid
- ! is the negation operator
- Multiple IP specifications can be included using square brackets [ ] and comma-separating. Do not add spaces

Snort Rule Header Port

```
                    Src IP                 Src Port          Dst IP             Dst Port
alert tcp $EXTERNAL_NET any -> $HOME_NET any
alert tcp 192.168.1.0/24 any -> 192.168.1.0/24 1:1024
alert tcp ![192.168.1.0/24,10.1.1.0/24] any -> 192.168.1.44
```

Port can be specified as:

- any -- any port
- 1:1024 -- ports 1 to 1024 inclusive
- 55: -- ports 55 and higher
- :55 -- ports 0 to 55 (inclusive)

negation still works:

- !6000:6001 - matches any port except 6000 and 6001

Snort Rule Header Direction

```
                 Src IP               Src Port      Dst IP        Dst Port
alert tcp $EXTERNAL_NET any -> $HOME_NET any
alert tcp 192.168.1.0/24 any -> 192.168.1.0/24 1:1024
alert tcp ![192.168.1.0/24,10.1.1.0/24] any -> 192.168.1.44
```

Direction can be specified as:

-> From right IP/Port (source) to left IP/Port (destination)

<> Any direction

Note: <- does not exist… so the snort rules always read consistently.

Snort Rule Option

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any \
 (msg:"SCAN SYN FIN";flags:SF; reference: arachnids,198;  \
classtype:attempted-recon; sid:624; rev:1;)
```

name:value;

msg: <sample message>	Logs message into /var/snort/log
flags: <AFPRSU210>	Matches specific TCP flags
content: <text>	Matches specified text in packet
content: |<hexadecimal>|	Matches specified hex chars
sid: <snort ID>	Unique number to identify rules easily. Your rules should use SIDs > 1,000,000
rev: <revision #>	Rule revision number
reference:<ref>	Where to get more info about the rule
gid:<generator ID>	Identifies which part of Snort generated the alert. See /etc/snort/gen-msg.map for values

Snort

- More in the lab 8 instruction!