

IDS

Carles Mateu

GREiA Research Group
Universitat de Lleida

IDS

Available IDS

Snort

OUTLINE

IDS

Available IDS

Snort

WHAT IS AN IDS/IPS?

An IDS is an **Intrusion Detection System**.

Any device/software monitoring either a *network* or *system(s)* to detect anomalous activity with *probably* malicious intent and inform the user (and log).

An IPS is an **Intrusion Prevention System**.

Any device/software monitoring either a *network* or *system(s)* to detect anomalous activity with *probably* malicious intent and avoid the event from taking place.

Many devices combine features from both IDS and IPS and Firewalls.

We can monitor lots of things:

- applications logs
- system state
- behaviour patterns
- network traffic

We can monitor lots of things:

- log of applications -> **log monitors**
- system state -> **integrity checkers**
- behaviour patterns -> **armoring**
- network traffic -> **sniffers**

An IDS combines most of these functions

- Distributed
 - Installed on hosts
 - Analyze local information
 - Detect problems
 - Communicate to coordinator there is a problem
- Centralized
 - Hosts send information to global coordinator
 - Coordinator analyzes all information
 - Detects problems

TYPES OF IDS (IPS)

- Host-based IDS (HIDS)
 - Usually Installed on servers (hosts)
 - Monitors local user activity
 - Monitors local program activity
 - Monitors system logs
- Network-based IDS (NIDS)
 - Sensors are installed *on the network*
 - Monitors anomalous network activity (deep packet inspection)
 - Logs/acts on that activity

Network IDS can be host focused also

- Firewalls -> door locks/doors/fence/gate
 - Do not react
 - **Prevent** intent
 - Medicine: gloves, hygiene, vaccines,...
 - *You lock your home door to prevent entry*
- IDS -> Alarm/Motion Detector
 - Dynamic and reactive
 - **React** on intent
 - Medicine: antivirals, antibiotics, ...
 - *You have a friend to come over trying to steal*

- Signature based IDS (rules)
 - Has database of “wrong” signatures, compares incoming traffic to those signatures
 - Most common used (we’ll use it)
- Anomaly based IDS
 - Learns the *right* behaviour of the system (network)
 - Generates alerts on deviating actions (strange packets)

- Anti-virus are signature based HIDS in fact.
- Pros:
 - Low false positives
 - Fast and known
- Cons:
 - Polymorphic payloads
 - *0-day* attacks

- AI fueled
- Pros:
 - Polymorphic payloads
 - *0-day* attacks
- Cons:
 - High false positives
 - Requires labeled data

We (everybody) will use traditional Machine Learning metrics to evaluate IDS:

- True Positives (TP): A real attack is detected
- True Negatives (TN): Good traffic is detected as good
- False Positives (FP): Regular traffic is tagged as an attack
- False Negatives (FN): Attacks are not detected

So, an IDS is:

- **Accurate:** if it detects all attacks
- **Precise:** if it never marks good traffic as an attack

We (everybody) will use traditional Machine Learning metrics to evaluate IDS:

- True Positives (TP): A real attack is detected
- True Negatives (TN): Good traffic is detected as good
- False Positives (FP): Regular traffic is tagged as an attack
- False Negatives (FN): Attacks are not detected

So, an IDS is:

- **Accurate:** if it detects all attacks
- **Precise:** if it never marks good traffic as an attack

We (everybody) will use traditional Machine Learning metrics to evaluate IDS:

- True Positives (TP): A real attack is detected
- True Negatives (TN): Good traffic is detected as good
- False Positives (FP): Regular traffic is tagged as an attack
- False Negatives (FN): Attacks are not detected

So, an IDS is:

- **Accurate:** if it detects all attacks
- **Precise:** if it never marks good traffic as an attack

We (everybody) will use traditional Machine Learning metrics to evaluate IDS:

- True Positives (TP): A real attack is detected
- True Negatives (TN): Good traffic is detected as good
- False Positives (FP): Regular traffic is tagged as an attack
- False Negatives (FN): Attacks are not detected

So, an IDS is:

- **Accurate:** if it detects all attacks
- **Precise:** if it never marks good traffic as an attack

We (everybody) will use traditional Machine Learning metrics to evaluate IDS:

- True Positives (TP): A real attack is detected
- True Negatives (TN): Good traffic is detected as good
- False Positives (FP): Regular traffic is tagged as an attack
- False Negatives (FN): Attacks are not detected

So, an IDS is:

- **Accurate:** if it detects all attacks
- **Precise:** if it never marks good traffic as an attack

We (everybody) will use traditional Machine Learning metrics to evaluate IDS:

- True Positives (TP): A real attack is detected
- True Negatives (TN): Good traffic is detected as good
- False Positives (FP): Regular traffic is tagged as an attack
- False Negatives (FN): Attacks are not detected

So, an IDS is:

- **Accurate:** if it detects all attacks
- **Precise:** if it never marks good traffic as an attack

- The true positive rate: $TP / (TP + FN)$
 - $TP + FN$ is the total number of attacks
- The false positive rate: $FP / (FP + TN)$
 - $FP + TN$ is the total size of good traffic

- Missing attacks lead to severe problems.
- False alarms lead to the system being ignored or disabled.

We **must** aim for accuracy **and** precision.

Suppose:

- 1% traffic are attacks
- 90% accuracy (TP rate)
- 10% FP rate

We detect an alarm: is it true? Probability of it being true?

- 1% traffic are attacks:
 - 1000 events: 990 good + 10 attacks
- 90% accuracy:
 - TPR: 90%, attacks: 10
 - $10 \times 90\% \rightarrow 9$ alarms generated by attacks
 - **1 attack goes undetected**
- 10% FP rate
 - FP: 10%, good events: 990
 - $990 \times 10\% \rightarrow 99$ false alarms
- $P(\text{attacks/alarms}) = 9/(9+99) = 0.08333$ (8.3%)

An alarm is probably a false alarm: **92% of the times!!!**

- 1% traffic are attacks:
 - 1000 events: 990 good + 10 attacks
- 90% accuracy:
 - TPR: 90%, attacks: 10
 - $10 \times 90\% \rightarrow 9$ alarms generated by attacks
 - **1 attack goes undetected**
- 10% FP rate
 - FP: 10%, good events: 990
 - $990 \times 10\% \rightarrow 99$ false alarms
- $P(\text{attacks/alarms}) = 9/(9+99) = 0.08333$ (8.3%)

An alarm is probably a false alarm: **92% of the times!!!**

- 1% traffic are attacks:
 - 1000 events: 990 good + 10 attacks
- 90% accuracy:
 - TPR: 90%, attacks: 10
 - $10 \times 90\% \rightarrow 9$ alarms generated by attacks
 - **1 attack goes undetected**
- 10% FP rate
 - FP: 10%, good events: 990
 - $990 \times 10\% \rightarrow 99$ false alarms
- $P(\text{attacks/alarms}) = 9/(9+99) = 0.08333$ (8.3%)

An alarm is probably a false alarm: **92% of the times!!!**

- 1% traffic are attacks:
 - 1000 events: 990 good + 10 attacks
- 90% accuracy:
 - TPR: 90%, attacks: 10
 - $10 \times 90\% \rightarrow 9$ alarms generated by attacks
 - **1 attack goes undetected**
- 10% FP rate
 - FP: 10%, good events: 990
 - $990 \times 10\% \rightarrow 99$ false alarms
- $P(\text{attacks/alarms}) = 9/(9+99) = 0.08333$ (8.3%)

An alarm is probably a false alarm: **92% of the times!!!**

- 1% traffic are attacks:
 - 1000 events: 990 good + 10 attacks
- 90% accuracy:
 - TPR: 90%, attacks: 10
 - $10 \times 90\% \rightarrow 9$ alarms generated by attacks
 - **1 attack goes undetected**
- 10% FP rate
 - FP: 10%, good events: 990
 - $990 \times 10\% \rightarrow 99$ false alarms
- $P(\text{attacks/alarms}) = 9/(9+99) = 0.08333$ (8.3%)

An alarm is probably a false alarm: **92% of the times!!!**

- Difficult tuning for “OK but not too much”
- Anomaly detectors may be trained by the attacker!
 - Attacker, over time, overwhelms normality
 - Train detector to attack-like behavior:
 - small doses
 - gradually over long periods of time
 - frog-boiling
- alarm behavior can itself be disruptive/taxing
 - may be used to deny service: log space filling, etc.
 - attacker could
 - trigger some alarm (lots of alarms)
 - use confusion for his real attack

- Difficult tuning for “OK but not too much”
- Anomaly detectors may be trained by the attacker!
 - Attacker, over time, overwhelms normality
 - Train detector to attack-like behavior:
 - small doses
 - gradually over long periods of time
 - frog-boiling
- alarm behavior can itself be disruptive/taxing
 - may be used to deny service: log space filling, etc.
 - attacker could
 - trigger some alarm (lots of alarms)
 - use confusion for his real attack

- Difficult tuning for “OK but not too much”
- Anomaly detectors may be trained by the attacker!
 - Attacker, over time, overwhelms normality
 - Train detector to attack-like behavior:
 - small doses
 - gradually over long periods of time
 - frog-boiling
- alarm behavior can itself be disruptive/taxing
 - may be used to deny service: log space filling, etc.
 - attacker could
 - trigger some alarm (lots of alarms)
 - use confusion for his real attack

- **Fragmentation:** split packets artificially
- **Spoofing:** alter TCP sequences
- **DoS:** raise tons of alarms
- **Train detectors:** generate over long periods of time alarms

- IDS' tasks are heavier than firewall's
- Should have access to lots of traffic
- Options:
 - Put between firewall/network (or router/network) -> will analyze all internet traffic
 - Use mirroring ports
 - Use several sniffers reporting back to central server

IDS

Available IDS

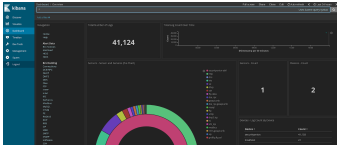
Snort



- Signature based IDS
- Can act as IPS
- Released in 1997, still actively maintained
- Rulesets freemium (free after 1 month)
- Huge community



- Signature based IDS
- Can act as IPS
- Released in 2009, still actively maintained
- Fast and modern architecture
- Shared tools with Snort



- Complete linux distro with several tools for IDS/analytics/etc.
- Includes Elasticsearch, Logstash, Kibana, Snort, Suricata, Sguil, Squert, etc.
- Requires min 4GB RAM....

OUTLINE

IDS

Available IDS

Snort

- We can use snort as:
 - packet sniffer (-v)
 - packet logger (-l)
 - network intrusion detector (-c)
- Files on:
 - /etc/snort/
 - snort.conf
 - rules/
 - /var/log/snort

```
alert tcp any any -> 193.144.10.50/32 80  
(flags:SF; msg: "files";)
```

Rule Header

```
alert tcp any any -> 193.144.10.50/32 80
```

Rule Options

```
(flags:SF; msg: "files";)
```

```
alert tcp any any -> 193.144.10.50/32 80  
(flags:SF; msg: "files";)
```

- alert: action
- tcp: protocol
- any any: source address and port
- -> : direction
- 193.144.10.50/32 80: destination address and port

SNORT RULES: HEADER: ACTION

```
alert tcp any any -> 193.144.10.50/32 80  
(flags:SF; msg: "files");
```

Available actions

- alert: Alerts and logs the packet when activated.
- log: Only logs the packet if activated.
- pass : Ignores or drops the packet and traffic matching.
- activate : Alerts and activates a dynamic rule(s).
- dynamic : Ignored, unless activated by the **activate** rule, then acts as a log rule.
- drop : blocks and logs the packet
- reject : blocks the packet, logs it, and send a TCP RST if protocol is TCP or an ICMP port unreachable if it's UDP.
- sdrops : block the packet but do not log it


```
alert tcp any any -> 193.144.10.50/32 80  
(flags:SF; msg: "files";)
```

We have available:

- tcp
- udp
- icmp
- ip

Work is on supporting: ARP, OSPF, RIP, etc.

SNORT RULES: HEADER: IP

```
alert tcp any any -> 193.144.10.50/32 80
alert tcp 193.144.12.0/24 any -> $INTERNAL_SERVER 80
alert tcp any any -> ![193.144.10.0/24,193.144.8.250] 80
```

- **any** is any IP address (0.0.0.0/0)
- IPs specified as CIDR notation.
- \$INTERNAL_SERVER defined in configuration (snort.conf)
- **!** is the negation
- Groups using `[]` and comma separated, without spaces!

SNORT RULES: HEADER: DIRECTION

```
alert tcp any any -> 193.144.10.50/32 80:83
alert tcp 193.144.12.0/24 any <> $INTERNAL_SERVER :30
alert tcp any !1024:65000 ->
    ![193.144.10.0/24,193.144.8.250] 80:
```

- -> from left (source IP and port) to right (destination IP and port)
- <> any direction (left to right or viceversa)
- <- *is not defined!!!!*

```
alert tcp any any -> 193.144.10.50/32 80  
(flags:SF; msg: "files";)
```

Format is: **name: value;**

- flags:SF; -> TCP flags to match (SYN FIN)
- msg: "files"; -> message to insert into alert

```
alert tcp any any -> 193.144.10.50/32 80  
(flags:SF; msg: "files"; sid: 1000001; rev:2; )
```

- sid:1000001; -> Rule ID, should be > 1000000
- rev: 1; -> Rule revision