Network Intrusion Detection and Prevention

- Motivation
- Basic Terms
- Common NIDS Deployment Scenarios
- Suricata NIDS
  - Basic Options
  - Modes
  - Configuration
  - Advanced Functionality
  - Rules

Motivation

- Firewall is an effective tool for filtering network traffic, network address translation (NAT), or “traffic shaping”
- Its disadvantage is that it does filter only based on the data in message headers, not taking into account the contents of the packet – application data
- Example:

```
# allow established connections
iptables -A INPUT -m state --state RELATED,ESTABLISHED \
-j ACCEPT
# allow connection requests to web server
iptables -A INPUT -p tcp -dport 80 -m state -state NEW \
-j ACCEPT
# drop everything else
iptables -A INPUT -j DROP
```

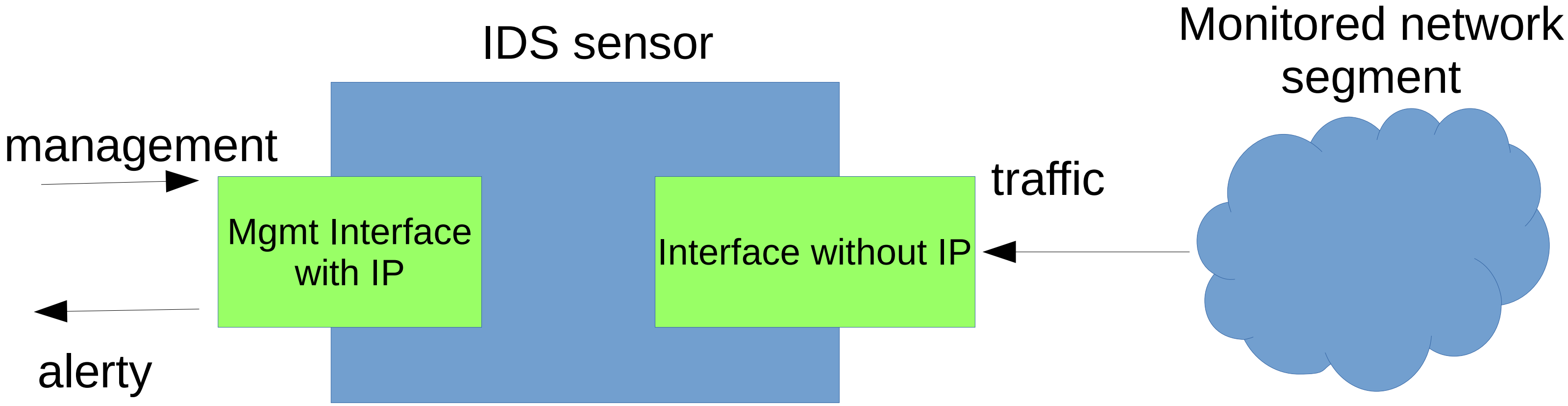
What happens, if a packet contains exploit for a vulnerability in a web server?

Basic Terms

- IDS (Intrusion Detection System) – monitors network traffic, analyses packet contents (headers + data), and in case the packet matches specific criteria it will create an alert for a security analyst. Security analyst has to constantly monitor the alerts.
- IPS (Intrusion Prevention System) – network traffic flows through it, analyses incoming packets, and drops those that match specific criteria. It requires detection criteria that are of high precision and quality (low False Positive rate).

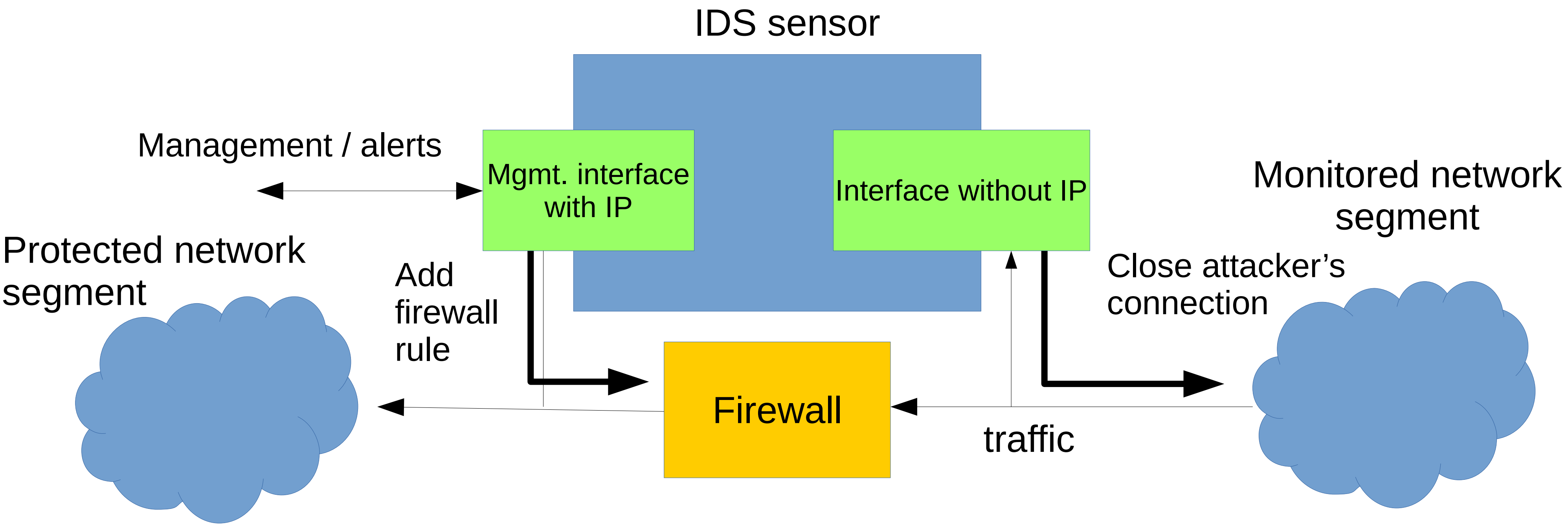
Deployment – IDS mode

- IDS mode – passive
  - Traffic monitoring
  - Connection via hub / span port / network tap



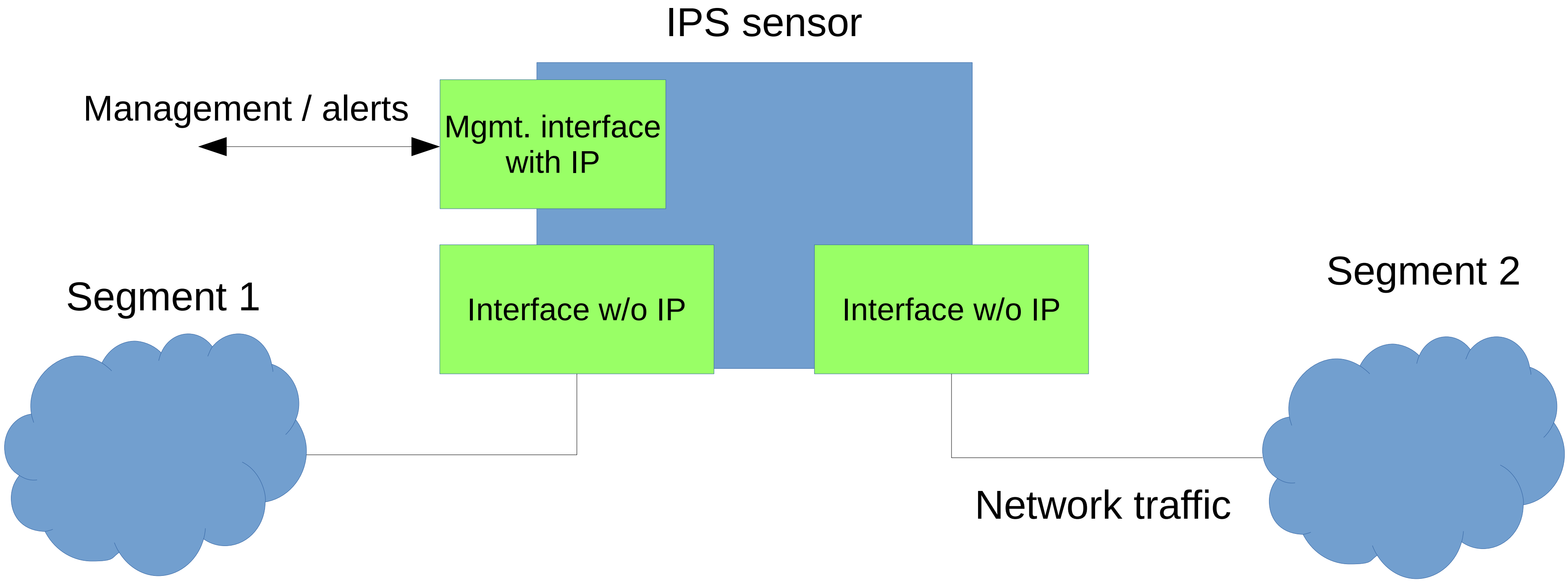
Deployment – IDS with Active Response

- IDS mode with active response
  - Traffic monitoring
  - Possibility to close attacker’s connection / edit firewall rules



Deployment – IPS mode

- IPS mode - inline
  - Traffic flows through the IPS sensor
  - If it matches a rule it is dropped (or logged)



Suricata

- “High performance Network IDS, IPS and Network Security Monitoring engine.”[1]
- Open Source.
- Developed by Open Information Security Foundation (OISF).
- Uses snort rule format.

[1] https://suricata.readthedocs.io/en/suricata-6.0.8/what-is-suricata.html

Suricata – Important Options

`-h` – print help message.

`-c <path>` – path to configuration file.

`-T` – test configuration.

`-v` – increase the verbosity of logging.

`-i` <interface> - analyse packets on network interface <interface>. This option will try to use the best capture method available.

`-l` <directory> - log alerts and packets to directory <directory> (default */var/log/suricata*).

`-k [all | none]` – Force (all) or disable (none) all checksum checks.

```
man suricata
```

Suricata – Runmodes (1)

- Two main modes
  - IDS – detection only,
  - IPS – detection and prevention.
- Each mode can further run in **one** of several *runmodes*.
- Runmode is a specific combination of
  - modules – parts of functionality, e.g. decode-module,
  - threads – module instances that process packets (multi-threading),
  - queues – passing a packet to the next thread.
- All runmodes have a name: single, workers, autofp.
- Usually tied to the choice of capture method

Suricata – Runmodes (2)

- PCAP_DEV (pcap live mode) – capture traffic using libpcap library. Suricata runs in IDS mode.
- PCAP_FILE (pcap file mode) – analyse PCAP file.
- AF_PACKET_DEV (af_packet IPS mode) – supports IPS/tap mode by bridging and monitoring network traffic between network interfaces.
- NFQ (local IPS mode) - capture traffic from iptables firewall using NFQUEUE. Suricata runs in IPS mode.
- Display available runmodes
  - `suricata --list-runmodes`

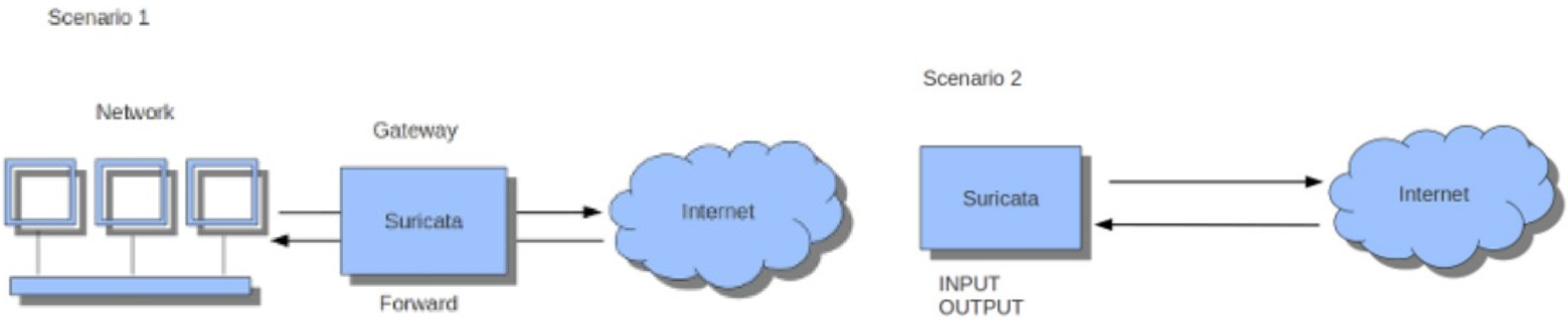
Suricata – IDS Mode

- Default mode.
- `–i` to select interface card you would like to sniff packets from.
- Suricata will try to use the available best capture method.
- Monitor traffic on interface eth0

  -
    ```
    suricata –c \
    /etc/suricata/suricata.yaml –i eth0
    ```

Suricata – IPS with Netfilter (1)

- **Layer 3 inline IPS mode**, suricata reads packets from iptables NFQUEUE (NFQ).
  - Scenario 1: protecting local network,
  - Scenario 2: protecting local host.



https://suricata.readthedocs.io/en/suricata-6.0.8/setting-up-ipsinline-for-linux.html#setting-up-ips-with-netfilter

Suricata – IPS with Netfilter (2)

- Monitored traffic from both directions has to be forwarded into the queue, (incl. RELATED,ESTABLISHED if used) otherwise session detection and communication may not work.
- Example: monitor HTTP traffic on port 80
  - `iptables –I INPUT –p tcp --dport 80 –j NFQUEUE`
  - `iptables –I OUTPUT –p tcp --sport 80 –j NFQUEUE`
  - `suricata –c /etc/suricata/suricata.yaml –q 0`
- When no userspace program is listening on a NFQ, all packets are queued and dropped.
  - --`bypass` switch changes this behaviour to ACCEPT for important traffic (such as SSH).

Suricata – AF_PACKET IPS mode

- **Layer 2 inline IPS mode**, suricata will copy packets between interfaces and act as an IPS between them.
- Section `af-packet` in configuration file.
- Two network interfaces required.
  - Both interfaces require enabled zero copy mode (`use-mmap: yes`).
  - Both interfaces require the same value of MTU.
- AF_PACKET capture method supports 2 modes
  - `copy-mode: ips` (drop keyword is used),
  - `copy-mode: tap` (no drop occurs).

Suricata – Configuration (1)

- Default configuration file */etc/suricata/suricata.yaml*
- Variables
  - `HOME_NET, EXTERNAL_NET`, etc. - home/protected network, external network
  - `HTTP_SERVERS, SMTP_SERVERS`, *etc.* - addresses of different servers on the network
  - `HTTP_PORTS, FTP_PORTS` – standard ports used by services on the network
- Outputs (logs)
  - *fast.log*, one line alerts (similar to snort fast.log).
  - *eve.log*, eve-log JSON format, useful when using suricata with other tools.
    - Use jq to view/filter alerts.
  - *suricata.log*, messages a suricata's functioning.
  - *stats.log*, statistic about traffic monitoring and filtering.
  - ...

Suricata – Configuration (2)

- Capture settings
  - Methods that will be used for capturing network traffic
- App Layer Protocol configuration
  - Application layer parsers/decoders
- Rule loading
  - By default load `suricata-update` managed rules.
  - Define additional `rule-files` for custom rules.
- Other settings
  - Detection settings, advanced settings, advanced traffic tracking, Defrag, ...

Suricata – Managing Suricata

- Manually
  - `sudo suricata –c <config> ...`
- As a service
  - `sudo systemctl start suricata`
  - Configuring startup parameters
    - */etc/sysconfig/suricata* (RedHat-based distros)
    - */etc/default/suricata* (Debian-based distros)

Suricata – Advanced Functionality

- Automatic protocol detection,
- Analysis of PCAP files,
- Protocol transactions,
- Network flows,
- PCAP recording,
- Extracted Files.

Suricata – Rules

- Defintions of malicious/suspicious content that is matched in network traffic (signatures).
- The most important part of IDS/IPS.
- Frequent updates required (multiple times / day).
- Free and commercial rule sets.
- Possibility to define custom rules.

Suricata – Managing Rules

- suricata-update rule management tool
- Download/update rules
  - `suricata-update`
  - By default uses Emerging Threats (ET) Open ruleset.
  - Default path to downloaded rules: */var/lib/suricata/rules.*
- Multiple rulesets available
  - `suricata-update list-sources`
  - `suricata-update enable-source <source>`
- Own rules
  - `local.rules`
  - restart to apply new rules.

Suricata – Rule Format

```
alert tcp $HOME_NET 2589 -> $EXTERNAL_NET any
(msg:"MALWARE-BACKDOOR..."; flow:to_client,established;
content:"2|00 00 00 06 00 00 00|Drives|24 00|"; depth:16;
metadata:ruleset community; classtype:misc-activity; sid:105; rev:14;)
```

- Action – after a match is found (alert, drop, reject, pass, etc.)
- Protocol – tcp, udp, ip, icmp, ftp, ssh, http, tls, ...
- IP address – variable, IP address, IP range (IP/mask, [IP1, IP2]), negation (!IP), any
- Port – variable, port, range, negation, any
- **Direction** – src **->** dst, **<>** two-way communication
- Options – message (being logged), search criteria, rule type, ID, ...

Suricata – Selected Rule Options

- `content –` Matching in packet contents
  - `content:“text”;` - text data
  - `content:”|0a0d|”;` - binary data
  - `content:”text”; nocase;` – case insensitive matching
  - `content:”/index.html”; http_uri;` – match on the request URI buffer
- `flow` – direction and state of the flow
  - state: established/not_established, only_stream/no_stream, only_frag/no_frag, ...
  - direction: to_server,to_client,from_server,from_client
  - `flow:established,to_server;`

Suricata – Selected Rule Options (cont.)

- `classtype` – classification of rules and alerts
  - based on definitions in *\/etc/suricata/classification.config*
  - `classtype:trojan-activity;`
- `sid: n;` – signature ID
  - rule ID, should be unique
- `rev: n;` – rule revision number

References

- man suricata
- https://suricata.io/
- https://suricata.readthedocs.io/en/suricata-6.0.8/
- https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Suricata_Developers_Guide