Network Intrusion Detection & Analysis

NIDS/NIPS & HIDS/HIPS

- Intrusion detection, prevention and analysis
- HID(P)S – host based intrusion detection(prevention) systems
- NID(P)S – network based intrusion(detection) systems
  - Functionality
  - Modes of detection
  - Types of NIDS/NIPS
  - Evidence acquisition
  - Packet logging
  - Systems – Snort (* *)

Firewalls and IPS/IDS

- IPS/IDS
  - typically designed to operate completely invisibly on a network.
  - respond directly to any traffic in a variety of ways.
    - (dropping packets, reseting connections, generating alerts, and even quarantining intruders).
    - may have the ability to implement firewall rules
- IPS technology
  - information on overly active hosts, bad logons, inappropriate content and many other network and application layer functions.

- Application firewalls
  - uses proxies to perform firewall access control for network and application-layer traffic.
  - Some have the ability to do some IPS-like functions,
    - RFC specifications on network traffic.
  - May provide real-time analysis and blocking of traffic.
  - Have IP addresses on their ports and are directly addressable.
  - Full proxy features to decode and reassemble packets.

Functionality

- IDS's are rule based
- Issues alerts
- Configured to capture suspicious packet sequences
- Sniffing
  - Multiple layer inspection
  - Protocol awareness
  - Protocol reassembly
- In a NIPS processing time is critical
- In a NIDS offline analysis and alerting is tolerable
  - Deep packet analysis is possible
- Some sort of normalization of packet contents may be required

Modes of Detection

- Signature based analysis
- Protocol analysis
- Behavioral analysis
- Active mode
- Passive Mode

Types of IDSs

- Commercial
  - Check Point IPS-1
    - http://www.checkpoint.com/products/ips-software-blade/
  - Cisco IPS
    - http://www.cisco.com/web/services/portfolio/product-technical-support/intrusion-prevention-ips/index.html
  - Enterasys IPS
    - https://www.enterasys.com/company/literature/ips-ds.pdf
  - Tipping Point IPS
    - http://h17007.www1.hp.com/us/en/whatsnew/040511-1.aspx

Types of IDSs

- Open Source
  - Snort
    - Snort can detect varied attacks like a buffer overflow, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, etc.
  - Security Onion
    - based on Ubuntu and comprises lots of IDS tools like Snort, Suricata, Bro, Sguil, Squert, Snorby, ELSA, Xplico, NetworkMiner, and many other
  - OpenWIPS-NG
    - OpenWIPS-NG is a free wireless intrusion detection and prevention system that relies on sensors, servers and interfaces.

http://opensourceforu.com/2017/04/best-open-source-network-intrusion-detection-tools/

Evidence Acquisition

- Types of evidence
  - Configuration
    - The configuration of each sensor is important
    - The location of each sensor within the network is also important
    - Running configuration is important
    - The rule set is important
  - Alert data
  - Packet header info
    - Flow data
  - Packet payloads
  - Correlation across multiple sensors

Configuration Files

- Alerts can be different on different sensors
  - The configuration of each sensor is important
  - The location of each sensor within the network is also important
  - Running configuration is important
  - The rule set is important

Comprehensive Logging

- All Packets all the time
  - Massive amounts of storage space
  - Difficult to archive except for NSA
  - Lots of CPU
  - Large risk
- Perhaps filter
- Only flow data

SNORT
https://www.snort.org/

- Most widely used IDS
- Libpcap utility
  - Functionality similar to tcpdump
- Layer 2 to Layer 4 analysis
- Capability to do Deep Packet Inspection
- Open rule language
- Extremely versatile
- Commercial Support
- Community/commercial business model

Architecture

- Uses libpcap to capture packet
- Passes through 4 preprocessors for reassembly and protocol analysis
  - Layer 3: reassembles fragments
  - Layer 4: reassembles streams
  - Layer 5: reassembles sessions
  - Layer 6: reassembles transactions
- Anamolies → alert at any layer
- After reassembly and anomaly detection
  - Information is handed off to rule engine
- Output engine then invoked for alerts
- Alert can be syslog / SNMP

Configuration is configurable

- /etc/snort/snort.conf
  - Global values of snort declared
  - Internal /external network definitions
  - Preprocessor configuration
  - Output processor configuration
  - Rule chunks
- /etc/snort/rules
  - Rules – each file can be enabled or disabled
- /var/log/snort
  - Native alerts – text based – corresponding packet captures

Rule Header

- Action
  - What must be done if a match in sensor (alert, log, pass, drop)
- Protocol
  - Protocol of the packet to match rule (tcp, udp, icmp, ip)
- Source IP/Network and port
- Directionality operator ← or →
- Destination IP / Network and port

Example

- `alert tcp any any → 192.168.2.1 80 (…)`
- `log udp 192.168.1.1 53 → !192.168.1.0/24 any (…)`
- `drop ip $EXTERNAL_NET any <> $HTTP_SERVERS $HTTP_PORTS (...)`

Rules in short

- The basis for logging or not logging a packet
- Can be more that one line long – now
  - Each line to be continued must be terminated with a `' \'`
    - » `That is "space \"`
- Generic syntax

```
rule_header (rule_options)
```

  - Rule header
    - » Action, addresses, ports, masks
  - Rule options
    - » Messages, what to look for, where to look

Rule Body

- General options (metadata about events)
  - provide a way of specifying information
- Detection options (stepwise instructions for matching packets or streams)
- Post-detection options (what to do if there is a match)

General Rule Options

- Msg : descriptive title to alerts
- Sid : Snort ID number uniquely identifying the rule
- Rev: Rule revision number
- Reference : optional pointer to background information URL etc.

Non Payload detection rule options

- Comparison operator for packet headers
  - TTL, IP options etc for IP, TCP header options, etc.

Payload detection rule options

- Content matching for ASCII string, Binary sequences
- Layer 7 Specific protocol data, such as HTTP, URIs, and SMTP commands
- Absolute and relational positional searches based on previous content match

Post Detection Rule Options

- translate rule matches into specific actions on a rule-by-rule basis, which then overrides the global Snort configuration
  - Causing alert in a different ways
  - Triggering capture of some portion of the packet
  - Response mechanisms such as reset of TCP connections etc.

Example

- `alert icmp $EXTERNAL_NET any -> $HOME_NET any ( msg :" ICMP PING "; icode :0; itype :8; classtype : misc - activity ; sid :384; rev :5;)`
- Alert on any inbound ICMP traffic that is of type 8 code 0: an “Echo Request.”

[**] [1:384:5] ICMP PING [**]

[Classification : Misc activity] [Priority : 3]

04/13 -03:12:08.359790 10.0.1.10 -> 10.0.1.254

ICMP TTL :64 TOS :0 x0 ID :38125 IpLen :20 DgmLen :84

Type :8 Code :0 ID :32335 Seq :1 ECHO

What does this do?

- `alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS ( msg :" WEB - MISC /etc / passwd "; flow : to_server , established ; content :"/ etc / passwd "; nocase ; classtype : attempted - recon ; sid :1122; rev :5;)`

[**] [1:1122:5] WEB - MISC / etc / passwd [**]
[Classification : Attempted Information Leak] [Priority : 2]
04/06 -05:11:46.015420 192.168.1.50:38097 -> 172.16.16.217:80
TCP TTL :64 TOS :0 x0 ID :9181 IpLen :20 DgmLen :165 DF
*** AP *** Seq : 0 x7D0FE4DE Ack : 0 x3EE535DC Win : 0 x5B4 TcpLen : 32
TCP Options (3) = > NOP NOP TS : 109823750 109820452

Conclusion

- Usually the trigger that launches an investigation
- The data for the investigation
- The proverbial haystack
- Snort often finds the needle