Chapter 7

Network Intrusion Detection and Analysis

2015. 11. 3

오 대 명(Daming Wu)

Email: wdm1517@gmail.com

Table of Contents

7.1 Why Investigate NIDS/NIPS?

- NIDS/NIPS alerts/logs may include details regarding illicit connections or even attempts. can provide a richer source of information.
- NIDS/NIPS can be configured to alert on, or at least log traffic that firewalls deem perfectly acceptable.
- An investigator could potentially modify a NIDS/NIPS configuration to begin detecting events it wasn’t previously configured to record.
- Rarely, the NIDS/NIPS itself might be suspected of compromise.

7.2 Typical NIDS/NIPS Functionality

NIDS/NIPS typically features include:

- **Rules:** Descriptions of how to compare a packet or stream with known malicious traffic.
- **Alerts:** Lists of suspicious packets/streams.
- **Packet captures:** Certain NIDS/NIPS can be configured to capture suspicious packets and save them for later analysis.

7.2.1 Sniffing

Passively: Connecting NIDS/NIPS to a mirroring port on a switch. Traffic is not detained.

Inline: Putting the NIDS/NIPS itself inline between two devices in a choke point position. Can cause noticeable latency.

7.2.2 Higher-Layer Protocol Awareness

- Protocol Reassembly

Attackers intentionally fragment the attack traffic. NIDS to perform fragment reassembly in order to be able to see and evaluate the target would receive and process.

- Normalization

HTTP protocol allows for many different ways of encoding data.

```
GET /../../../../etc/passwd HTTP/1.1
GET %2f..%2f..%2f..%2f..%2fetc%2fpasswd HTTP/1.1
GET %2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2f%65%74%63%2f%70%61%73%73%77%64
    HTTP/1.1
```

  - write a “signature” to detect each and every one of the various ways such a string can be represented.
  - normalize each string to its “canonical” form, and only then compare them against the list of “known bad” things.

7.2.3 Alerting on Suspicious Bits

NIDS/NIPS can be configured to communicate alerts via various means. The most common of these include:

- Sending email alerts
- Logging events to a syslog server
- Sending SNMP traps
- Logging events directly to a queriable database
- Store alert and event data locally

7.3 Modes of Detection

- Signature-Based Analysis:

This method compares headers, contents of packets, and streams of packets against databases of known, malicious byte sequences in order to identify suspicious traffic.

- Protocol Awareness:

Protocol-aware NIDS/NIPS reassemble fragments (Layer 3), reassemble streams (Layer 4), and even reconstruct entire protocols (Layer 7) because they have to understand how the endpoint of the communication will interpret the data.

- Behavioral Analysis:

The idea behind this approach is to spend time building a “baseline” of normal network behavior against which to compare future behavior. “learns” what is normal, in order to later detect aberrations.

7.4 Types of NIDS/NIPSs

- Commercial
  - Check Point IPS-1
  - Cisco IPS4
  - Corero Network Security5
  - Enterasys IPS6
- Roll-Your-Own
  - It is both free for use under the GNU Public License (GPL)
  - Supported by commercially funded research and development efforts.

7.5 NIDS/NIPS Evidence Acquisition

Types of Evidence

- Configuration
  - NIDS/NIPS has not been configured to alert upon a particular event, then the absence of alerts for event is meaningless.
- Alert data
  - Evaluate network traffic based on a set of rules or learned patterns, identify traffic of interest, and produce alerts.
- Packet header and/or flow record information
  - Log packet header and/or flow record data. help identify the origin, destination, and patterns of activity.

7.5 NIDS/NIPS Evidence Acquisition

Types of Evidence

- Packet payloads
  - NIDS/NIPS capture full packet contents from packets that trigger alerts, and subsequent packets.
- Activities correlated across multiple sensors
  - Topologically and temporally correlated event data is exceptionally useful.

7.5.2 NIDS/NIPS Interfaces

NIDS/NIPS are almost uniformly designed to be information aggregation devices, so they are typically deployed with a central analysis console.

- GUI Interfaces
  - accessed, inspected, and configured via some sort of graphical user interface.
  - web client, special client
- CLI Interfaces
  - via SSH or direct console connection and use the command-line interface to view configuration, logs, and alerts.
- Off-System Logging
  - NIDS/NIPS systems are capable of sending critical evidence to an aggregation repository.

7.6 Comprehensive Packet Logging

- Full content packet logging is occurring on NIDS/NIPS.
- Only packets being captured and made available for future inspection are trigger alerts.
- NIDS/NIPS generally are not configured to capture the packets that preceded an incident, nor the ones that followed.
- Problem for the investigator: often very little can be gleaned from a single packet, as too much of the context is unavailable.

7.7 Snort

An overview of Snort:

- Most widely used NIDS
- Open-source code
- Open rule language
- Extremely versatile
- Actively improving, partly due to commercial support

7.7.1 Basic Architecture

- Snort pulls packets in using libpcap, from whichever interface is specified.
- Snort passes all packets through preprocessors for reassembly and protocol analysis.
  - At Layer 3, it reassembles fragments
  - At Layer 4, it reassembles streams
  - At Layer 5, it reassembles sessions
  - At Layer 7, it reassembles transactions
- If at any of those layers, Snort detects anomalies, it can alert.
- After the analyzed information is handed off to the Snort rule engine. This engine can then use any and all of the protocol information and payload contents to detect malicious traffic.
  - The output engine alerts will be communicated to the end-user.

7.7.2 Configuration

- **/etc/snort/snort.conf:** global values for Snort are declared, internal/external network definitions, preprocessors will be configured, output processors will be configured for logging, and major of rules.
- **/etc/snort/rules/:** This directory contains the rules files themselves. Within each file, rules can be disabled or enabled.
- **/var/log/snort/:** This directory contains the native alerts file, where Snort records text-based alerts, and the libpcap files with corresponding packet captures.

7.7.3 Snort Rule Language

Rule Header

- Action: This field describes what the Snort sensor should do when a packet is discovered to match the rule. Typical actions are alert, log, pass, and drop.
- Protocol: This field describes the protocol the packet must be employing to match the rule. Possible values are icmp, tcp, udp, or the general case, ip.
- Source IP/network and port: These fields describe what the source of a packet must be match.
- Directionality operator: This field allows the rule author to specify whether a match one direction only (from source to destination) or bidirectionally. The allowed values are “->” and “<>”.
- Destination IP/network and port: These fields describe what the destination of a packet must match.

7.7.3 Snort Rule Language

Examples

- 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 (...)

7.7.3.2 Rule Body

General Rule Options (metadata about events) include:

• msg: Descriptive title that will be attached to any resulting alerts.

• sid: The required Snort ID number identifies the rule.

• rev: The rule's revision number. The combination of "sid" and "rev" define rules uniquely.

• reference: An optional pointer to background information, often a CVE number, or URL.

7.7.3.2 Rule Body

Nonpayload Detection Rule Options

• Comparison operators for all of the protocol fields in the:

– IP packet header (TTL, fragmentation information, embedded protocol, IP options, etc.)

– TCP segment header (TCP flags and stateful flow inspection, sequence, and acknowledgment numbers, window size, etc.)

– ICMP header (ICMP type and code, as well as sequence values)

7.7.3.2 Rule Body

Payload Detection Rule Options

- Content matching for:
  - ASCII strings
  - Binary sequences
  - PCREs
- Layer 7-specific protocol data, such as HTTP URIs and SMTP commands
- Absolute- and relative-positional searches, based on previous content matches

7.7.3.2 Rule Body

Post-Detection Rule Options

• Causing the alert and packet logging for the given rule to be handled in a different way than other alerts.

• Triggering the capture of some or all of the subsequent packets in a stream after a single packet has caused a rule to fire.

• Active response mechanisms, including the bidirectional reset of TCP connections and the sending of ICMP Destination Unreachable packets.

7.7.4 Examples

- **Snort Rule**

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg :" ICMP PING ";

icode :0; itype :8; classtype :misc - activity ; sid :384; rev :5;)

- **Snort Packet**

03:12:08.359790 IP 10.0.1.10 > 10.0.1.254: ICMP echo request , id 32335 ,

seq 0, length 64

- **Snort Alert**

  [**] [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

7.8 Conclusion

- we reviewed typical NIDS/NIPS functionality, including sniffing and higher-layer protocol analysis.
- We discussed modes of detection, different types of NIDS/NIPS, and the various types of evidence that can be recovered from them.
- We examined Snort, a popular open-source NIDS/NIPS, and concluded with examples of Snort rules, captured packets, and alerts.

Thanks!