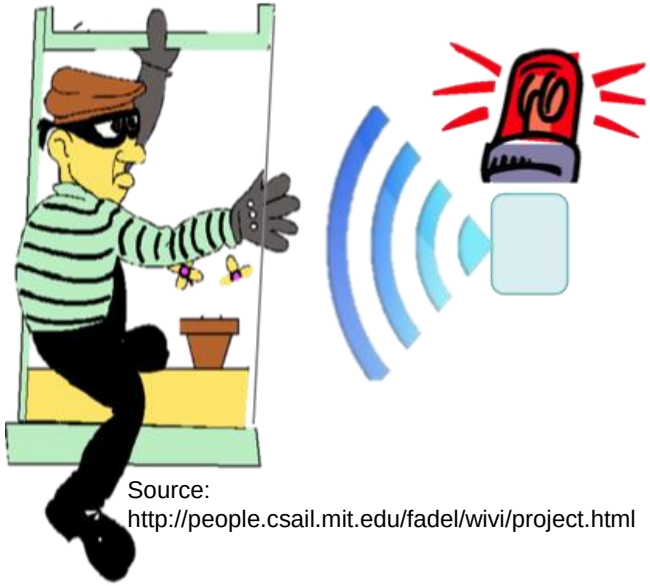
Network Security (NetSec)

Summer 2015
Exercise 4: Intrusion Detection Systems



Source:
http://people.csail.mit.edu/fadel/wivi/project.html

Prof. Dr.-Ing. Matthias Hollick

Technische Universität Darmstadt
Secure Mobile Networking Lab - SEEMOO
Department of Computer Science
Center for Advanced Security Research Darmstadt - CASED

Mornewegstr. 32
D-64293 Darmstadt, Germany
Tel.+49 6151 16-70922, Fax. +49 6151 16-70921
http://seemoo.de or http://www.seemoo.tu-darmstadt.de

Milan Schmittner
milan.schmittner@seemoo.tu-darmstadt.de

Learning Objectives

Discuss a component to implement network security:

- Understand purpose, role, and architecture of Intrusion Detection Systems (IDS)
- Investigate IDS concepts and working in detail
- Exemplify operation of IDS with Snort
- Discuss limitations of IDS and evasion techniques

Motivation for Intrusion Detection Systems (IDS)

SECURE MOBILE NETWORKING

Intrusions are not unique to networks

- Can you give examples of "IDS"s in real life?

Why IDS in networks?

Definitions & Elements of IDS

SECURE MOBILE NETWORKING

Intrusion

- A set of actions aimed to compromise the security goals, namely integrity, confidentiality, availability,… of a computing and networking resource

Assumptions

- System activities are observable
- Normal and intrusive activities have distinct evidence

Intrusion detection

- The process of identifying (and responding) to intrusion activities
- Components of intrusion detection systems
  - Algorithmic perspective vs. from a system architecture perspective

IDS Components

Algorithmic perspective

- Features: capture evidences extracted from audit data
- Modeling–analysis approach: piecing the evidences together
  - Signature-based detection (identify misuse)
  - Statistical anomaly-based detection (identify anomalies)

System architecture perspective

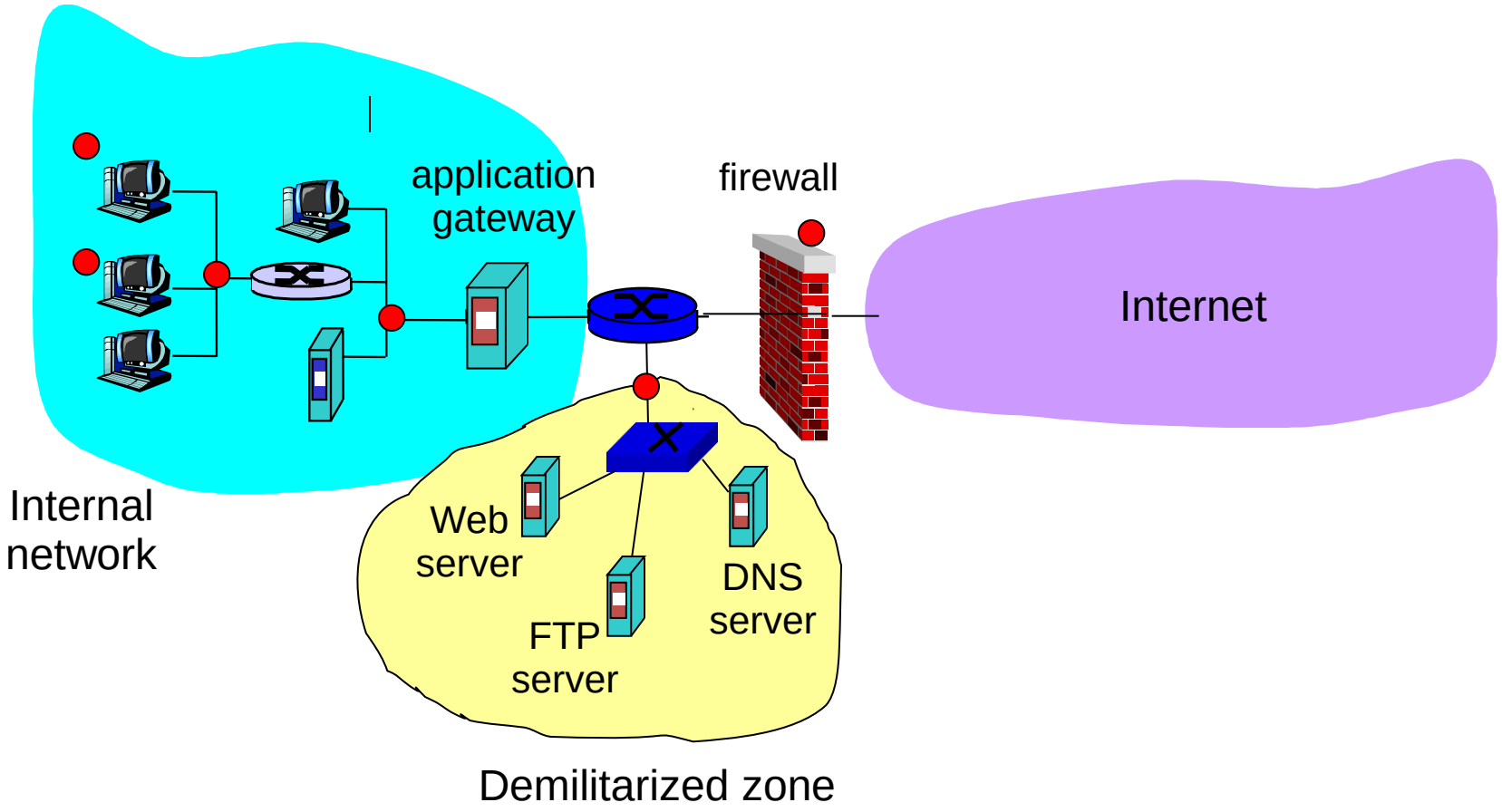
- Deployment: network based or host based
- Components: audit data processor, knowledge base, decision engine, alarm generation and responses

Development and maintenance

- Hand-coding of "expert knowledge"
- Learning based on audit data

IDS Sensor Placement

Distinction into host-based and network-based IDS



Key Performance Metrics

Accuracy and *effectiveness*:

few false positives and false negatives desirable

- Alarm: A
- Intrusion: I

	A	¬A
I	True positive	False negative
¬I	False Positive	True negative

Performance: the rate at which traffic/audit events are processed

- Hardware-based solutions, architecture decisions (e.g. place multiple IDSs downstream instead of central location)

Resilience and fault tolerance: resistance to attacks

- Should be run on a single hardened host that supports only intrusion detection services

Timeliness: time elapsed between intrusion and detection

Signature-based Detection

Sniff traffic on network

- border router or multiple sensors within a LAN

Match sniffed traffic with signatures

- attack signatures in database

Signature: set of rules pertaining to a typical intrusion activity

- Example: any ICMP packet > 10,000 bytes
- Example: more than one thousand SYN packets to different ports on same host under a second -> possible port scan

Warn administrator when signature matches

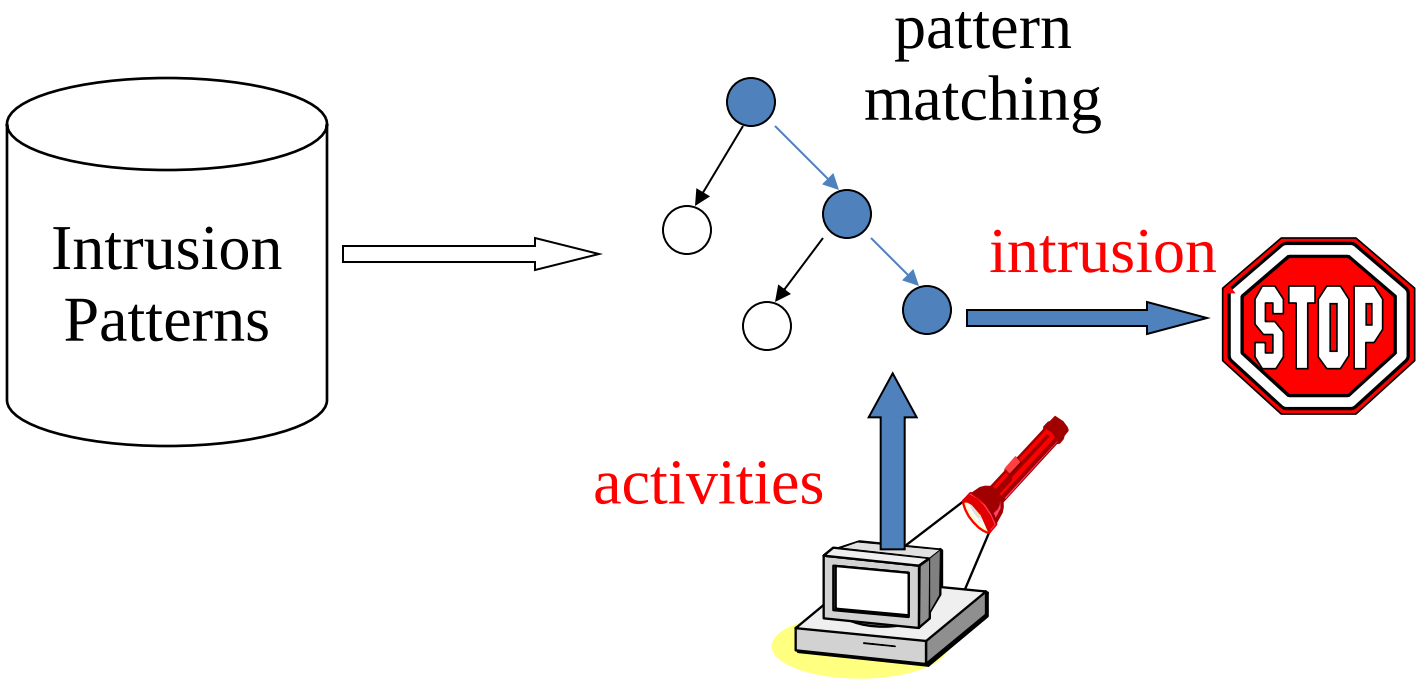
But how to derive rules?

Signature-based Detection

Rule-based penetration identification

- uses expert system technology
- with rules identifying known penetration, weakness patterns, or suspicious behavior
- compare audit records or states against rules
- rules usually machine & O/S specific
- rules are generated by security experts/engineers/admins who interview & codify knowledge of security admins
- quality depends on skill and thoroughness of those involved setting up the system

Signature-based Detection



Example: ***if*** (src_ip == dst_ip) ***then*** "LAND attack"

Problems?

Can't detect new/unknown attacks

Limitations of Signature-based Detection

SECURE MOBILE NETWORKING

Requires previous knowledge of attack to generate accurate signature

- Blind to unknown attacks

Signature bases are getting larger

- Every packet must be compared with each signature
- IDS can get overwhelmed with processing; can miss packets

→ Still, predominant approach to NIDS

Statistical Anomaly-based Detection

SECURE MOBILE NETWORKING

Observe traffic during normal operation
- Create normal traffic profile

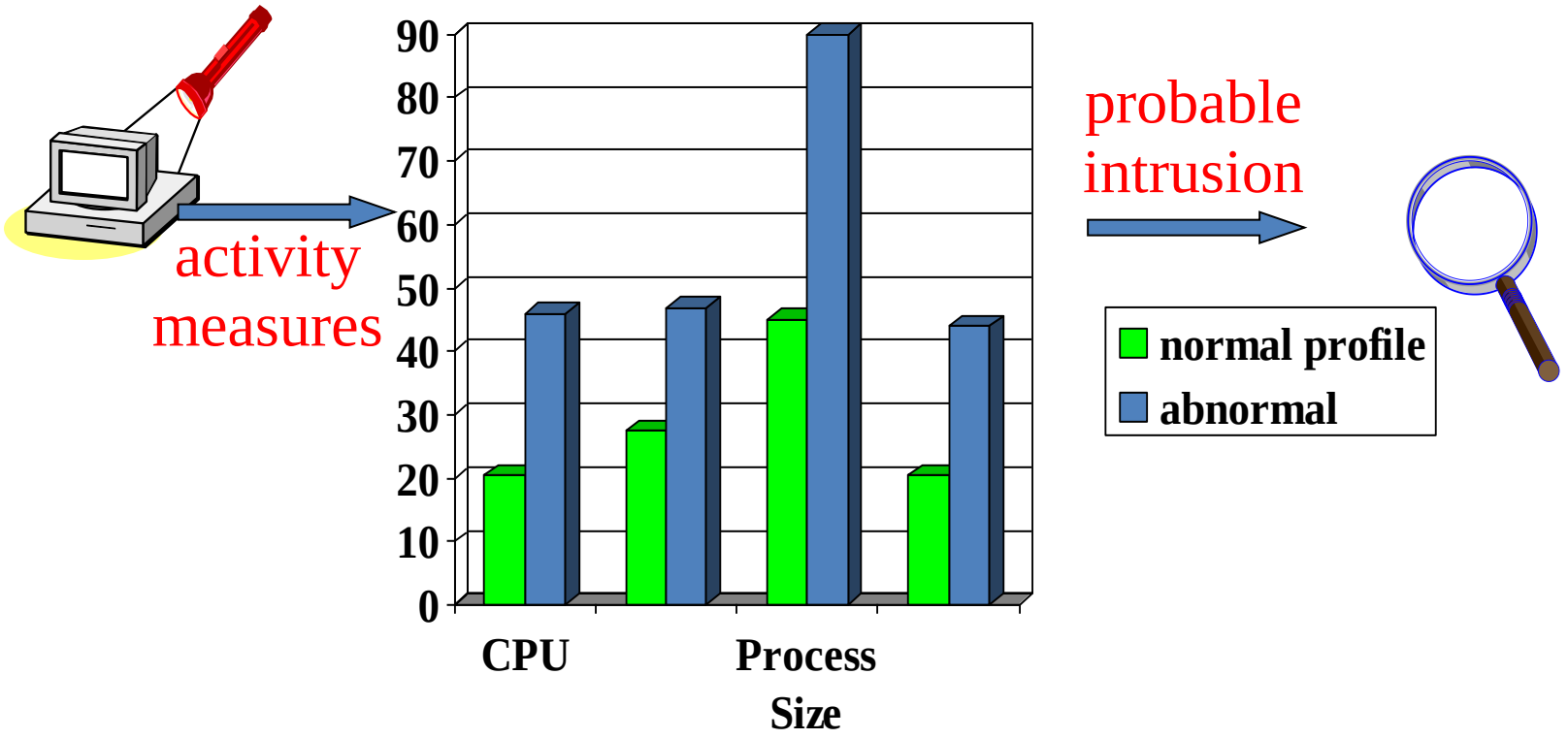
Look for packet streams that are statistically unusual, e.g.,
- inordinate percentage of ICMP packet
- or exponential growth in port scans/sweeps

Doesn't rely on having previous knowledge of attack

Threshold detection
- count occurrences of specific event over time
- if exceed reasonable value assume intrusion
- by itself crude and ineffective

Profile-based detection
- characterize past behavior of users
- detect significant deviations from this
- profile usually based on multiple parameter

Limitations of Statistical Anomaly-based Detection



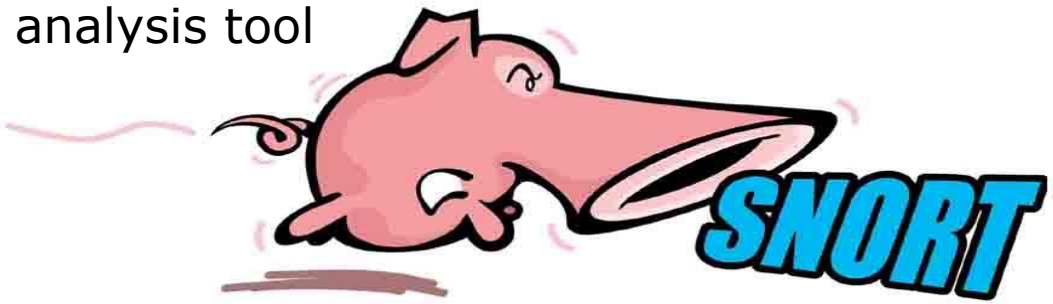
Relatively high false positive rate – anomalies can just be the "new" normal.

Intro to Snort

SECURE MOBILE NETWORKING

What is Snort?

- Snort is a multi-mode packet analysis tool



Where did it come from?

- Developed out of the evolving need to perform network traffic analysis in both real-time and for forensic post processing

- Portable (Linux, Windows, MacOS X, Solaris, BSD, IRIX, Tru64, HP-UX, etc.) and fast
- Configurable (Easy rules language, many reporting/logging options)
- Free (GPL/Open Source Software)

Snort Design

SECURE MOBILE NETWORKING

Overview

- Packet sniffing "lightweight" network intrusion detection system
- Libpcap-based sniffing interface
- Plug-in system allows for flexibility
  - Plug-ins for preprocessor, detection, output

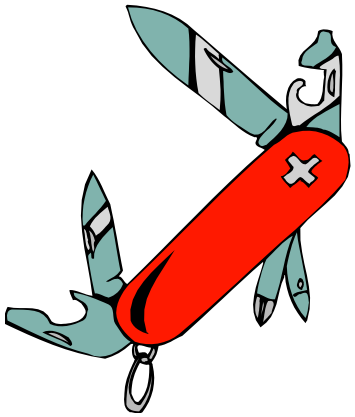
Detection Engine

- Signature-based
- Modular detection elements are combined to form these signatures
- Wide range of detection capabilities
  - Stealth scans, OS fingerprinting, buffer overflows, back doors, CGI exploits, etc.
- Rules system is very flexible, and creation of new rules is relatively easy

Using Snort

Three main operational modes

- Sniffer Mode
- Packet Logger Mode
- **NIDS Mode**
- (Forensic Data Analysis Mode)



Operational modes are configured via command line switches

- Snort automatically tries to go into NIDS mode if no command line switches are given, looks for snort.conf configuration file in /etc

Snort Rule Example

Sample rule to detect SubSeven trojan:

```
alert tcp $EXTERNAL_NET 27374 -> $HOME_NET any
  (msg:"BACKDOOR subseven 22"; flags: A+; content:
  "|0d0a5b52504c5d3030320d0a|"; reference:arachnids,485;
  reference:url,www.hackfix.org/subseven/; sid:103;
  classtype:misc-activity; rev:4;)
```

Elements before parentheses comprise 'rule header'
Elements in parentheses are 'rule options'

Basic Configuration: snort.conf

Defining some useful variables:

```
var HOME_NET 193.152.1.1/24
var EXTERNAL_NET !193.152.1.1/24
var HTTP_SERVERS 193.152.1.17
var HTTP_PORTS 80 8080
```

Snort Rules

```
alert tcp $EXTERNAL_NET 27374 -> $HOME_NET any
  (msg:"BACKDOOR subseven 22"; flags: A+; content:
  "|0d0a5b52504c5d3030320d0a|"; reference:arachnids,485;
  reference:url,www.hackfix.org/subseven/; sid:103;
  classtype:misc-activity; rev:4;)
```

alert action to take; also **log**, **pass**, **activate**, **dynamic**

tcp protocol; also **udp**, **icmp**, **ip**

$EXTERNAL_NET source address; this is a variable – specific IP is ok

27374 source port; also **any**, negation (**!21**), range (**1:1024**)

-> direction; best not to change this, although **<>** is allowed

$HOME_NET destination address; this is also a variable here

any destination port

Snort Rules

```
alert tcp $EXTERNAL_NET 27374 -> $HOME_NET any
  (msg:"BACKDOOR subseven 22"; flags: A+; content:
  "|0d0a5b52504c5d3030320d0a|"; reference:arachnids,485;
  reference:url,www.hackfix.org/subseven/; sid:103;
  classtype:misc-activity; rev:4;)
```

msg:"BACKDOOR subseven 22"; message to appear in logs

flags: A+; TCP flags; many options, like `SA`, `SA+`, `!R`, `SF*`

content: "|0d0…0a|"; binary data to check in packet; content without | (pipe) characters do simple content matches

reference…; where to go to look for background on this rule

sid:103; rule identifier (**mandatory**)

classtype: misc-activity; rule type; many others

rev:4; rule revision number

other rule options possible, like **offset, depth, nocase**

More Snort Rules Examples

Rules which actually caught intrusions

- `alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 1433 (msg:"MS-SQL xp_cmdshell - program execution"; content: "x|00|p|00|_|00|c|00|m|00|d|00|s|00|h|00|e|00|l|00|l|00|"; nocase; flags:A+; classtype:attempted-user; sid:687; rev:3;)`
  caught compromise of Microsoft SQL Server
- `alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS cmd.exe access"; flags: A+; content:"cmd.exe"; nocase; classtype:web-application-attack; sid:1002; rev:2;)`
  caught Code Red infection
- `alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"INFO FTP \"MKD / \" possible warez site"; flags: A+; content:"MKD / "; nocase; depth: 6; classtype:misc-activity; sid:554; rev:3;)`
  caught anonymous ftp server

Snort – Conclusion

Snort is a powerful tool, but maximizing its usefulness requires a trained operator

Becoming proficient with network intrusion detection takes 12 months; “expert” 24-36?
(says the author of Snort)

Snort is considered a superior NIDS when compared to most commercial systems

Summary and Cautious Note

SECURE MOBILE NETWORKING

Summary

- Discussed principles and practice on NIDS

However

- Signature-based
  - "We have the largest knowledge/signature base"
  - But ineffective against new attacks, no prediction, static
- Statistical anomaly-based
  - "x% detection rate and y% false alarm rate"
  - Creates computational overhead and might miss damaging intrusions?

Also: successful intrusion detection depends on policy, management as much as technology

- Security Policy (defining what is acceptable and what is being defended) is the first step
- Notification (who, how fast)?
- Response Coordination

Even More Snort Rule Examples

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any
(msg:"ICMP PING NMAP";dsize:0;itype:8;sid:10;rev:1;)
```

- Rule generates alert for ICMP having empty payload, ICMP type 8, and
  arriving from the outside.
- This is part of an NMAP ping.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 139
(msg: "DOS SMBdie attack":; flags: A+;
 content:"|57724c6568004577a|"; sid:11; rev:1;)
```

- Rule generates alert if a TCP packet from outside contains `|57724c6568004577a|` in payload and is headed to port 139 (netbios) for some internal host.
- This is part of a buffer overflow attack on a computer running Server Message Block Service.

Even More Snort Rule Examples

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
(msg:"WEB-IIS ISAPI .ida attempt"; uricontent:".ida?";
nocase; dsize:>239; flags:A+; sid:12; rev:1)
```

- Rule generates alert for packet heading to Web server with .ida? in URL in GET message
- Buffer overflow attack that allows attacker to take over server.

Snort Rule Writing

SECURE MOBILE NETWORKING

Example: Cross-site scripting (XSS):

Web site allows scripts to be inserted into dynamically created Web page. Can reek havoc.

Look out for HTTP requests containing <SCRIPT>

Might first try:

- `alert tcp any any -> any any (content: "<SCRIPT>"; msg: "XSS attempt"; sid:14; rev:1)`
- triggers many false positives: e.g., e-mail message with JavaScript

Then try:

- `alert tcp $EX_NET any -> $HTTP_SRVS $HTTP_PRTS (content: "<SCRIPT>"; msg: "XSS attempt"; nocase; sid:14; rev:2)`

Snort Rule Syntax

Rule is a single line
- Rule header: everything before parenthesis
- Rule option: what's in the parenthesis

Syntax for rule header:

```
rule_action protocol src_add_range src_prt_range
dir_operator dest_add_range dest_prt_range
```

Example:

```
alert tcp 192.168.1/24 1:1024 -> 124.17.8.1 80
```

rule actions: alert, log, drop
protocol: tcp, udp, icmp
direction: -> and <>
src, dest port ranges :

Snort Rule Syntax (2)

SECURE MOBILE NETWORKING

Syntax for rule option:

One or more option keywords

- separated by semi-colons

Example:

- `(msg: "XSS attempt"; content: "<SCRIPT>"; nocase; sid:14; rev:2)`

Content-related keyword examples:

content: "smtp v2"; (ascii)
content: "|0f 65 a7 7b|" ; (binary)
uricontent: ".ida?";
content-list: "inappropriate_content.txt";
nocase;
offset: 20; (start at byte 20 in payload)
depth: 124; (stop at byte 124 in payload)

Snort Rule Syntax (3)

IP-related keyword examples:

ttl: <5;

id:2345; (id field, used for fragments)

fragoffset: 0;

dsize: >500; (payload size)

ip_proto: 7;

ICMP-relayed keyword examples:

itype: 8;

icode: 3;

Snort Rule Syntax (4)

TCP-related rules

flags: A+; (ACK flag)

flags: FUP; (FIN, Urgent, or Push flag)

- + alert if specified bit is discovered, in addition to at least one other
- ! alert if any of the specified bits is *not* set

seq: 12345432; ack: 54321234;

Response examples

msg: "christmas tree attack";

logto: "new_rule.log"; logs packet when match occurs