

# Intrusion Detection Systems @ unfug

Christian "Rancor" Fischer, Timo "bluec0re" Schmid,  
Sergej "winnie" Schmidt

9. Juni 2011

# Inhaltsverzeichnis

## 1 Intrusion Detection Systems

- What and Why
- NIDS/HIDS
- IPS
- Rules

## 2 Snort

- Some details...
- Using Snort as an IPS
- Analysis

## 3 Snort Rules

- Writing your own
- Header
- Options
- Rule Example

# Intrusion Detection Systems

## what is that?

- detect break-in
- analysing/protocolling attack

## Why

- firewall failed
- attack from inside

->no protection & no protocolling



# Network/Host

- Network Detection Intusion Systems
- Host Detection Intrusion Systems

# IPS

- Intrusion Prevention System

# Rules

- less is more
- false positive alarm are evil

## Some details..

### ...to snort

- First version released in 1998 by Martin Roesch
- Now developed by Sourcefire
- GNU GPL and commercial version available
- The "de facto standard for IPS"

## Some details..

### ...how snort works

- three main modes: sniffer, packet logger and network intrusion detection
- Checks traffic against rulesets
- Output alerts: syslog, binary tcpdump format or into a database

# Preprocessors

## What are preprocessors?

- Drop modular plugins into Snort
- Code is run before the detection engine is called, but after the packet has been decoded
- Many available: SSH, DNS, SSL/TLS, ARP Spoof

# Rules

## How to get up2date rules?

From <http://www.snort.org>

- With paid subscription directly after release
- As a registered user 30-days after initial release to subscribers

## Alternatives

Get free, more up2date rules:

- <http://www.bleedingsnort.com/downloads/bleeding.rules.tar.gz>
- <http://www.emergingthreats.net/>
- Using a the perl script <http://oinkmaster.sourceforge.net>

# Inline mode

Not enabled by default

## How to configure

- Start Snort with: `snort -Q`
- Config option: `config policy_mode:inline`
- Test first with: `snort -enable-inline-test`

# Snortsam

3rd Party plugin Snortsam → <http://www.snortsam.net/>

## Some details

- Supports: ipchains, iptables, ebtables, pf and many more
- One or more agents are running on firewall
- Snort sensor is configured with the address of the agent
- Extending the rules that should request a blocking action with certain parameters

# Analyze alerts using BASE

## Basic Analysis and Security Engine

- <http://base.secureideas.net/>
- Needs a webserver with php
- ->Live-Demo

# Writing your own Snort Rules

## Rules?

- simple, lightweight description language
- used to identify possible attacks/floods/etc.

## Layout

Divided into to logical sections:

- Rule Header
- Rule Options

# Rule Header

Rule header specifies which packets should be investigated by this rule and which action should be executed

## Layout

- 1 action  
 $\in \{alert, log, pass, activate, dynamic, drop, reject, sdrop\}$
- 2 protocol  $\in \{TCP, UDP, ICMP, IP\}$
- 3 source ip and netmask (! prefix for negation)
- 4 source port (! prefix for negation)
- 5 direction operator  $\in \{->, <>\}$
- 6 destination ip and netmask (! prefix for negation)
- 7 destination port (! prefix for negation)

## Rule Header - action

### action

- alert  $\Rightarrow$  generate an alert and log the packet
- log  $\Rightarrow$  log the packet
- pass  $\Rightarrow$  ignore the packet
- activate  $\Rightarrow$  alert and turn on another dynamic rule
- dynamic  $\Rightarrow$  wait for activation, then act as log rule
- drop  $\Rightarrow$  drop and log the packet
- reject  $\Rightarrow$  drop, log the packet and sends an protocol reject
- sdrop  $\Rightarrow$  drop the packet

## Rule Header - Examples

### Examples

```
alert tcp any any -> any any
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS
    $HTTP_PORTS
alert icmp $EXTERNAL_NET any -> $HOME_NET any
alert udp $EXTERNAL_NET any -> $HOME_NET 9
alert tcp $EXTERNAL_NET any <> $HOME_NET 179
```

# Rule Options

## Categories

- general ⇒ informations about the rule
- payload ⇒ looks inside the packet payload
- non-payload ⇒ looks outside the packet payload
- post-detection ⇒ triggers after a rule has fired

## Rule Options - Categories

### General

Keyword	Description
msg	message for alert and log
reference	reference to external systems
gid	generator id; identifies the part of Snort which generates the event
sid	unique id of rule
rev	unique revision of rule
classtype	categorizes the attack
priority	assigns severity level
metadata	key/value pair for additional informations

## Rule Options - Categories

### Payload I

Keyword	Description
content	search for content
rawbytes	search for bytes, ignoring encoding
depth	depth to search in packet
offset	start point
distance	how many bytes should be ignored after last match
within	
uricontent	
isdataat	verifies that the payload has data at a specified location

## Rule Options - Categories

### Payload II

Keyword	Description
pcre	perl compatible regex
byte_test	test byte field against a value
byte_jump	read length from data and skip it
ftpbounce	detects FTP bounce attacks
asn1	decodes a packet and looks for malicious encodings
cvs	detects invalid entry strings
dce_iface / dce_opnum / dce_stub_data	DCE/RPC 2 Preprocessor

## Rule Options - Categories

### Non-Payload I

Keyword	Description
fragoffset	compare IP fragment offset against a value
ttl	compare IP ttl against a value
tos	compare IP tos against a value
id	compare IP id against a value
ipopts	check for present IP options
fragbits	check for fragmentation and reserved bits set
dsize	test packet payload size
flags	check TCP flags
flow	specifies traffic flow direction
flowbits	track states during TCP session

## Rule Options - Categories

### Non-Payload II

Keyword	Description
seq / ack	check TCP sequence / acknowledge number
window	check TCP window size
itype / icode	ICMP type / code
icmp_id / icmp_seq	ICMP id / sequence value
rpc	check for RCP applications, version and procedure number in SUNRPC CALL
ip_proto	check for IP protocol header
same_ip	check if src and dst IPs are the same

## Rule Options - Categories

### Post-Detection I

Keyword	Description
logto	log to a special output file
session	extract user data from TCP sessions
resp	close session if alert is triggered
react	react to traffic by closing connection and sending notice
tag	log more than just the packet which had triggered the rule
activates	add a rule if a specific network event occurs

## Rule Options - Categories

### Post-Detection II

Keyword	Description
activated_by	enable a rule when an activate rule was triggered
count	used with <i>activated_by</i> , specifies how many packets should be left for the following rule
replace	replace the matched content with string of same length
detection_filter	specifies a treshold when an event will be generated



## Rule Options - Examples

### Examples

```
(msg:"COMMUNITY ICMP undefined code"; icode
  :>18; classtype:misc-activity; sid
  :100000197; rev:1;)
(msg:"TFTP Get"; content:"|00 01|"; depth:2;
  classtype:bad-unknown; sid:1444; rev:3;)
```

# Rule Example

## Simple rule

- should investigate packets on http
- should be triggered if someone sends the string "c99.php"

## Rule Example

```
1 alert tcp any any -> any 80 \  
2   (content:"c99.php"; reference:mcafee, 136948; \  
3   msg:"Request for C99-Shell"; \  
4   sid:12345678; rev:1; classtype:misc-attack;)
```

## Rule Example

```
1 alert tcp any any -> any 80 \  
2     (content:"c99.php"; reference:mcafee, 136948; \  
3     msg:"Request for C99-Shell"; \  
4     sid:12345678; rev:1; classtype:misc-attack;)
```

## Rule Example

```
1 alert tcp any any -> any 80 \  
2     (content:"c99.php"; reference:mcafee, 136948; \  
3     msg:"Request for C99-Shell"; \  
4     sid:12345678; rev:1; classtype:misc-attack;)
```

## Rule Example

```
1 alert tcp any any -> any 80 \  
2     (content:"c99.php"; reference:mcafee, 136948; \  
3     msg:"Request for C99-Shell"; \  
4     sid:12345678; rev:1; classtype:misc-attack;)
```