IDS / SNORT

Matsuzaki ‘maz’ Yoshinobu

<maz@iij.ad.jp>

stole slides from

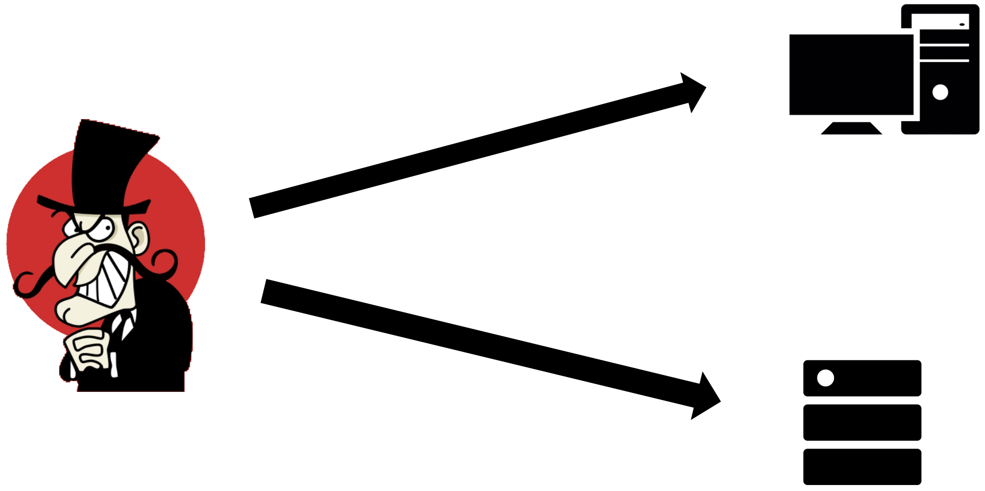
Fakrul Alam

Sometimes, Defenses Fail

- Our defenses aren’t perfect
  - Patches weren’t applied promptly enough
  - Antivirus signatures not up to date
  - 0-days get through
  - Someone brings in an infected USB drive
  - An insider misbehaves
- Now what?
- Most penetrations are never detected
  - This allows continuing abuse, and helps the attackers spread elsewhere

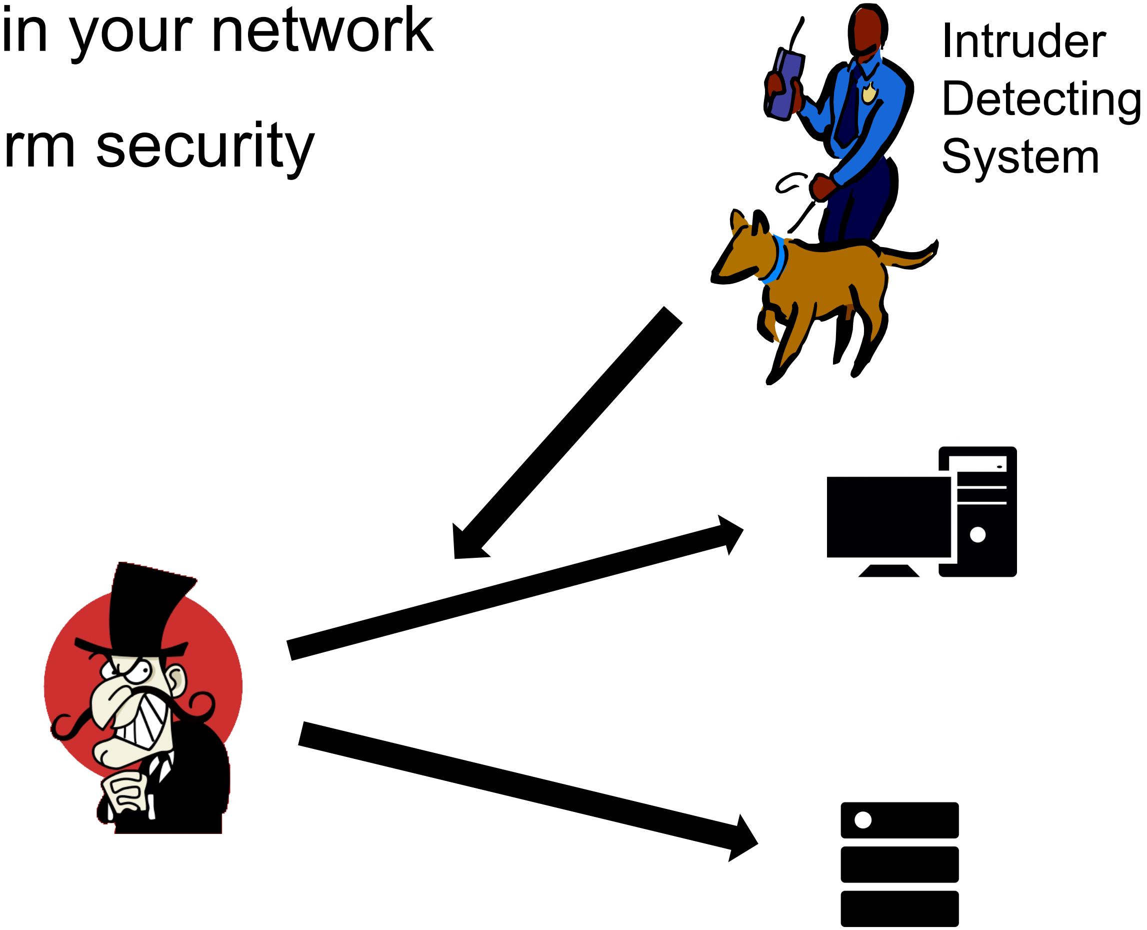
Unexpected Activity

- There could be an intruder even if you have security practice in place



Additional Monitoring

- Activity in your network
- To confirm security



Intruder Detecting System

What can IDS realistically do

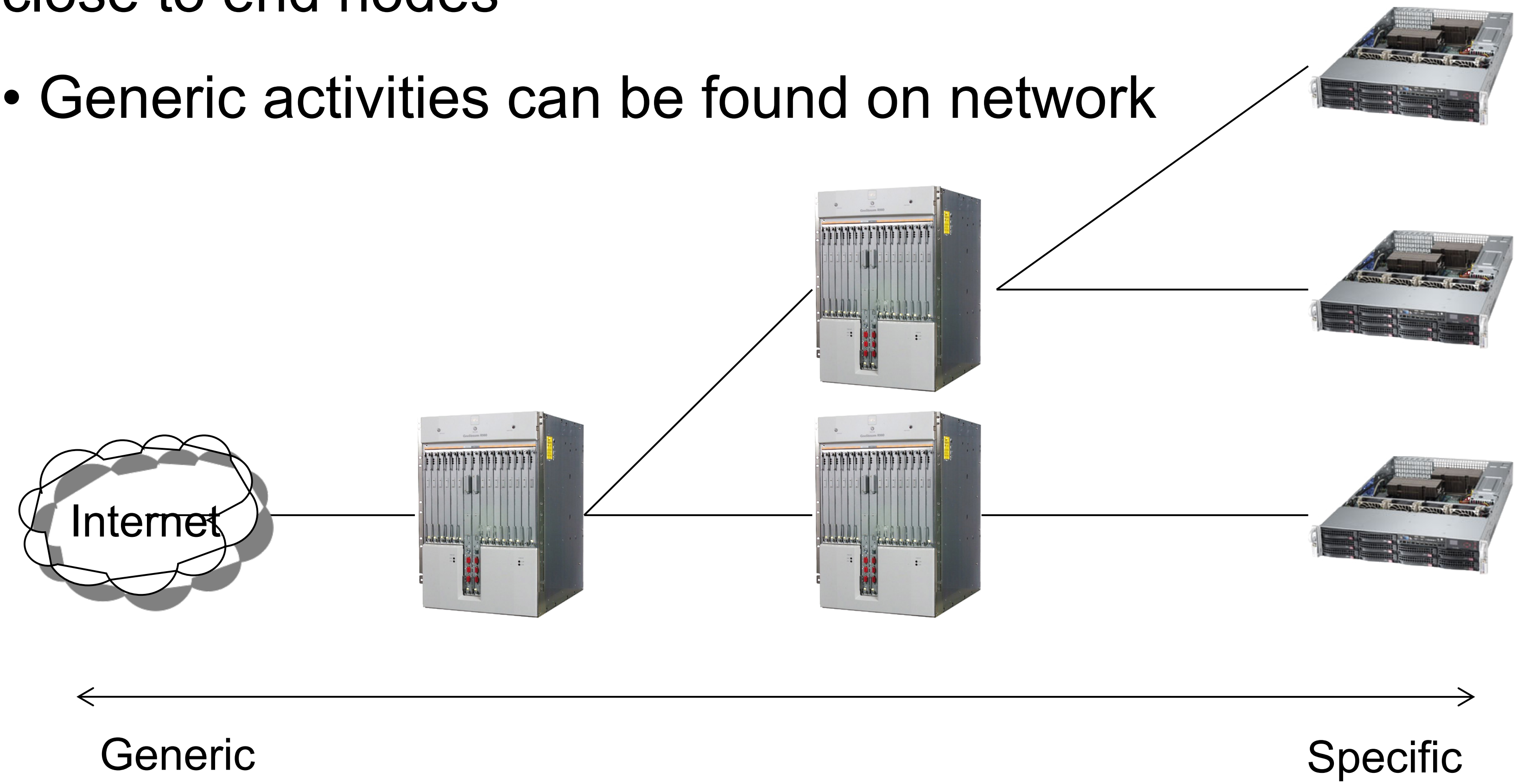
- Detect successful attacks
- Look for various things that shouldn’t be there
- Infected files
- Attacks on other machines
- Packets that shouldn’t exist
- Strange patterns of behavior
- Contain attacks before they spread further
- Clean up penetrated machines—because you’ll know they’re infected
- Recognition of pattern reflecting known attacks
- Statistical analysis for abnormal activities

What IDS can’t do

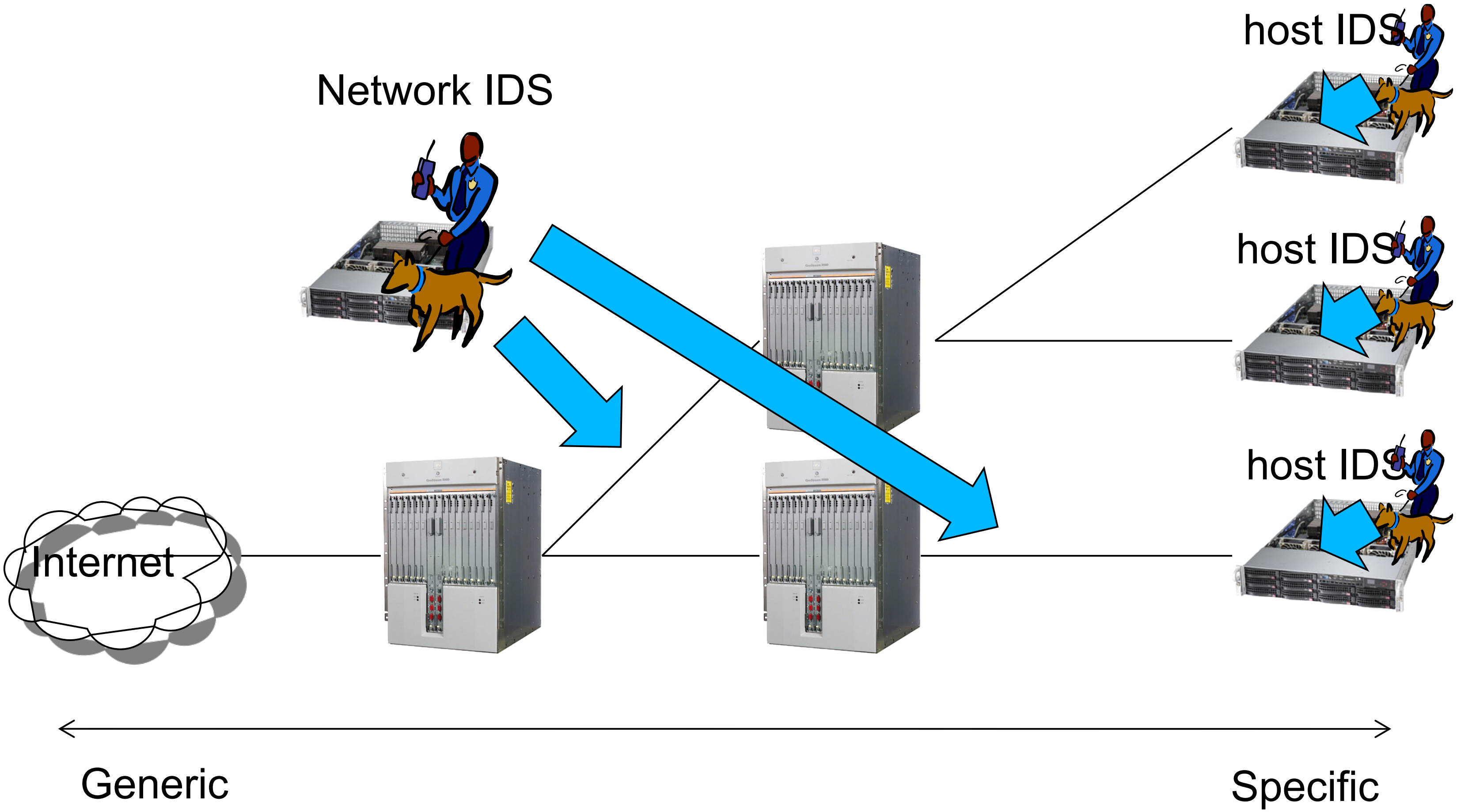
- Compensate for weak authentication & identification mechanisms
- Investigate attacks without human intervention
- Guess the content of your organization security policy
- Compensate for weakness in networking protocols, for example IP Spoofing

Monitoring Point

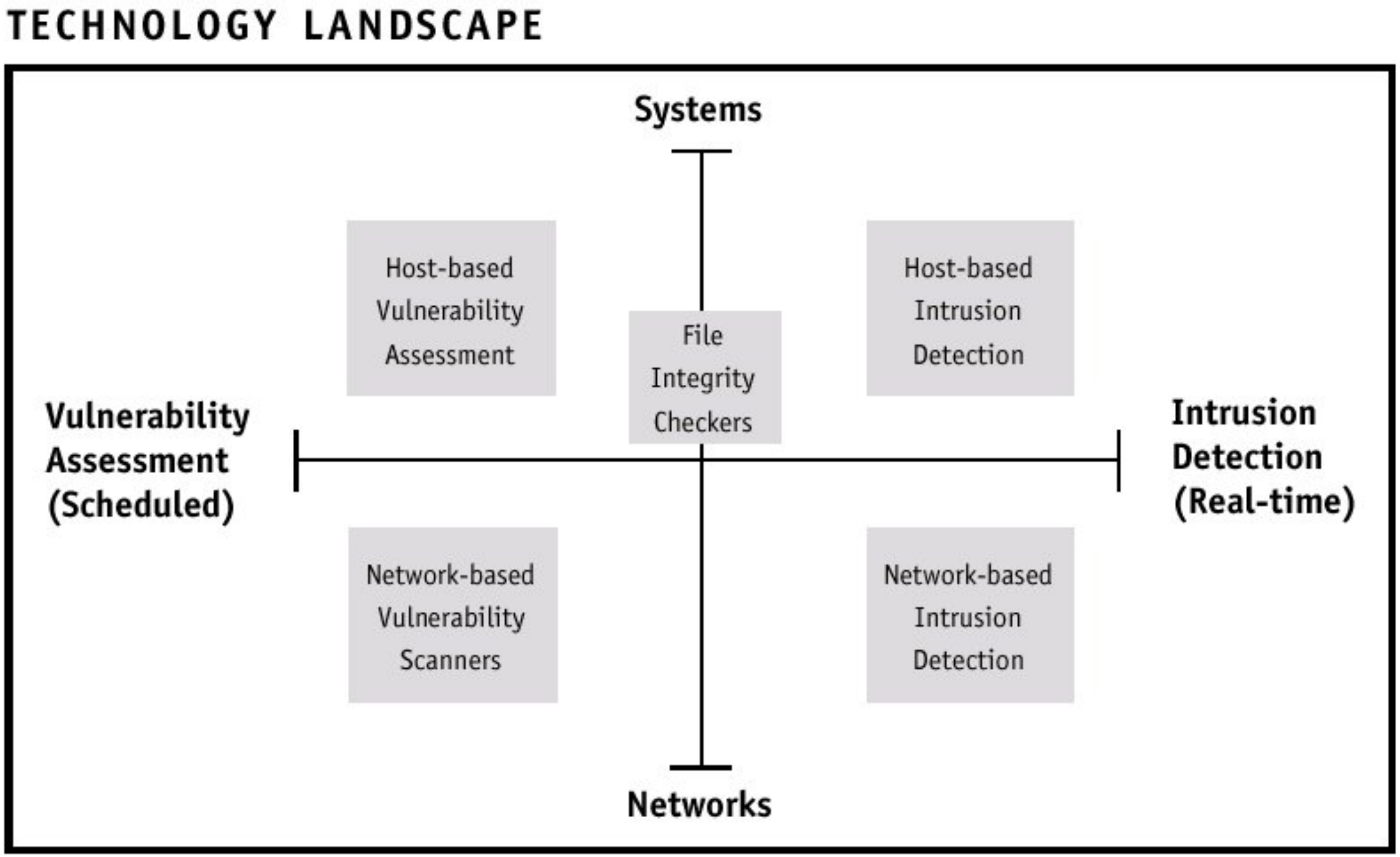
- More specific rules can be applied for a point close to end nodes
- Generic activities can be found on network



Network and Host IDS



IDS Technology landscape


TECHNOLOGY LANDSCAPE
Systems
Host-based Vulnerability Assessment
File Integrity Checkers
Host-based Intrusion Detection
Vulnerability Assessment (Scheduled)
Intrusion Detection (Real-time)
Network-based Vulnerability Scanners
Network-based Intrusion Detection
Networks

Preventive

Real Time

Alert

- You may receive tons of millions of alerts
  - –Depending on your detection rules
  - –There are many suspicious activities in the Internet today
- You should notice a critical one at least
  - –Detection rule is important!

Alert

- False Positive / Type I Error:
  - –is the incorrect rejection of a true null hypothesis
  - –is when a system raises an incorrect alert
- False Negative / Type II Error:
  - –is the failure to reject a false null hypothesis
  - –is when an attack pass undetected

Types of Detection

- Signature Based
  - –Match patterns against known attacks
  - –Catch the intrusions in terms of the characteristics of known attacks or system vulnerabilities
- Anomaly Based
  - –Look for unusual behavior
  - –Detect any action that significantly deviates from the normal behavior

Intrusion Detection for ISPs

- Monitor your own network—but that's no different than any other enterprise
- Monitor your customers
  - Good: you can help them by detecting problems
  - Good: you can prevent them from clogging your infrastructure
  - Bad: it can be privacy-invasive

Open Source IDS/IPS

- SNORT
- Security Onion
- OSSEC
- OpenWIPS-NG
- Suricata
- Bro IDS

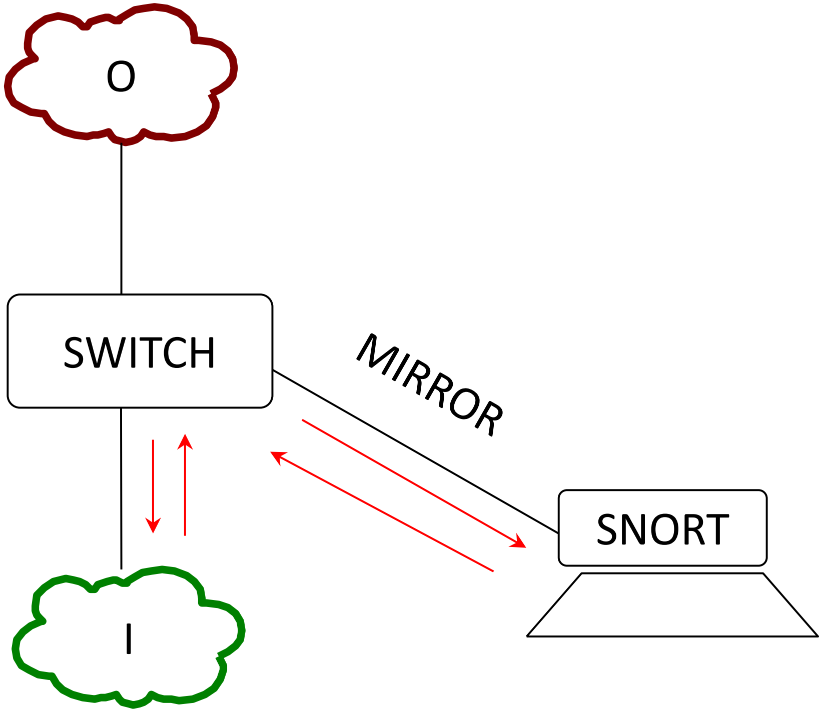
SNORT

- Snort is an open source IDS, and one of the oldest ones
- Hundreds of thousands of users
- Active development of rules by the community make Snort up to date, and often more so than commercial alternatives
- Snort is fast! It can run at Gbit/s rates with the right hardware and proper tuning

Getting Snort to see the network

- You could run Snort in multiple ways
  - As a device "in line" behind or after the firewall/router
    - But this adds one more element that can fail in your connectivity
  - Or you could use a span/mirror port to send traffic to Snort
  - Or you can use an "optical splitter" to "mirror" or "tap into" traffic from a fiber optic link
    - This method and the previous are the most recommended

Getting Snort to see the network



Getting Snort to see the network

- Be careful not to overload your switch port – If you mirror a gigabit port to another gigabit port, the monitoring port (the receiving port) can drop packets if the total traffic exceeds 1 Gbit/s

Monitoring Port…

- On Cisco Catalyst, this is a “SPAN” port
- You can SPAN one port to another, a group of ports to one port, or an entire VLAN to a port
- Sample config:

```
interface FastEthernet 0/1
# port monitor FastEthernet 0/2
```

- This would copy any packet received on F0/2 to F0/1

Snort configuration file

- By default, /etc/snort/snort.conf
- It’s a long file – 900+ lines
- If you browse it, you will notice many “preprocessor” entries
- Snort has a number of “preprocessors” which will analyze the network traffic and possibly clean it up before passing it to the rules

SNORT Rules

- Snort rules are plain text files
- Adding new rules to snort is as simple as dropping the files into /etc/snort/rules/
- Groups of rules can be loaded from snort.conf using the “include” statement
- Rules can match anything
- Technical – web attacks, buffer overflow, portscan, etc…
- Policy/user oriented – URL filtering, keyword, forbidden applications, etc…

Tailoring the rules

- Not all rules will make sense in your network
- You will want to customize which rules you want to run
- Otherwise you will get many false positives, which will lead you to ignore Snort, or simply turn it of…
- It doesn’t help to have logs full of junk alerts you don’t want
- To avoid this, rules can be suppressed (disabled)

Updating Snort rules

- The commercially maintained snort rules are available for free with a 30 day delay from http://www.snort.org/start/rules
- Other rules are maintained by some volunteers at emerging threats: http://rules.emergingthreats.net/open/
- The updating of rules can be automated with a tool called “Pulled Pork”, which is located at http://code.google.com/p/pulledpork/

Snort rules

- Snort rules are divided into two logical sections:
  - –**Rule Header** : The rule header contains the rule's action, protocol, source and destination IP addresses and netmasks, and the source and destination ports information.
  - –**Rule Options** : The rule option section contains alert messages and information on which other parts of the packet should be inspected to determine if the rule action should be taken.

Snort rules

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 22
(msg: "SSH Detected"; sid:10; rev:1;)
```

The text up to the first parenthesis is the rule header and the section enclosed in parenthesis contains the rule options. The words before the colons in the rule options section are called option *keywords*.

sid: snort rule identifier
rev: snort rule revision identifier

Snort rules header

- **alert** - generate an alert using the selected alert method, and then log the packet
- **log** - log the packet
- **pass** - ignore the packet
- **activate** - alert and then turn on another dynamic rule
- **dynamic** - remain idle until activated by an activate rule , then act as a log rule
- **drop** - block and log the packet
- **reject** - block the packet, log it, and then send a TCP reset if the protocol is TCP or an ICMP port unreachable message if the protocol is UDP.
- **sdrop** - block the packet but do not log it. (silently drop)

Snort rules : The Direction Operator

- The direction operator **->** indicates the orientation, or direction, of the traffic that the rule applies to.
- There is no **<-** operator.
- Bidirectional operator **<>**

Snort rules : sid

- The sid keyword is used to add a “Snort ID” to rules
  - Range 0-99 is reserved for future use
  - Range 100-1,000,000 is reserved for rules that come with Snort distribution
  - All numbers above 1,000,000 can be used for local rules

Snort rules : classtype

- Rules can be assigned classifications and priority numbers to group and distinguish them
  - `/etc/snort/classification.config`

```
config classification: DoS, Denial of Service Attack,2
```

Name Description Priority

- You can distinguish between high- and low-risk alerts

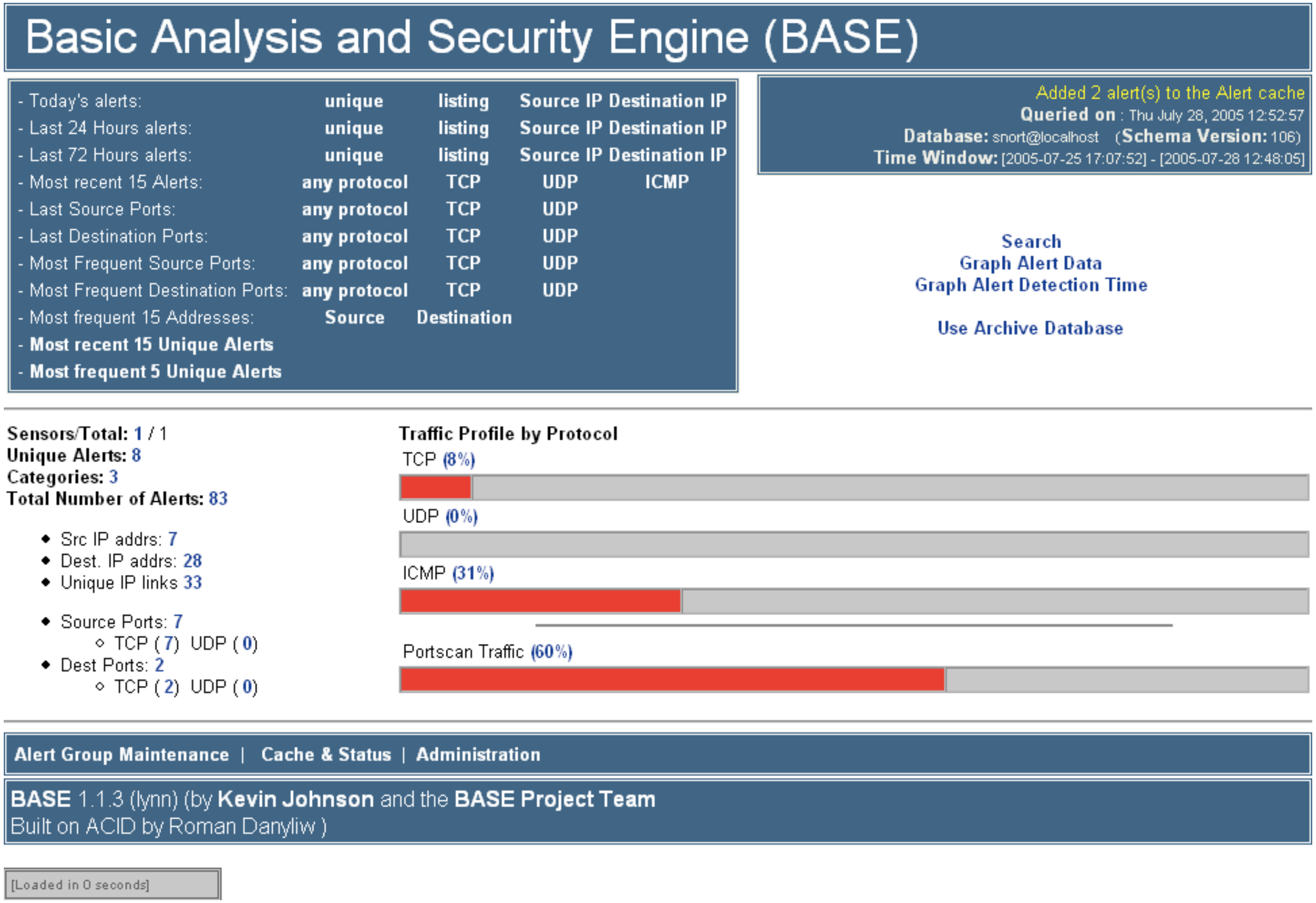
Sample rules

```
alert tcp msg:"MYSQL root login attempt";
flow:to_server,established; content:"|0A 00 00 01 85 04 00 00
80|root|00|"; classtype:protocol-command-decode; sid:1775;
rev:2;)

alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 3306 (msg:"MYSQL
show databases attempt"; flow:to_server,established;
content:"|0F 00 00 00 03|show databases"; classtype:protocol-
command-decode; sid:1776; rev:2;)

alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 3306 (msg:"MYSQL
4.0 root login attempt"; flow:to_server,established;
content:"|01|"; within:1; distance:3; content:"root|00|";
within:5; distance:5; nocase; classtype:protocol-command-
decode; sid:3456; rev:2;)
```

Reporting and logging

- Snort can be made to log alerts to an SQL database, for easier searching
- A web front-end for Snort, BASE, allows one to browse security alerts graphically

BASE (Basic Analysis and Security Engine)



Basic Analysis and Security Engine (BASE)

- Today's alerts:	unique	listing	Source IP	Destination IP
- Last 24 Hours alerts:	unique	listing	Source IP	Destination IP
- Last 72 Hours alerts:	unique	listing	Source IP	Destination IP
- Most recent 15 Alerts:	any protocol	TCP	UDP	ICMP
- Last Source Ports:	any protocol	TCP	UDP	
- Last Destination Ports:	any protocol	TCP	UDP	
- Most Frequent Source Ports:	any protocol	TCP	UDP	
- Most Frequent Destination Ports:	any protocol	TCP	UDP	
- Most frequent 15 Addresses:	Source	Destination		
- **Most recent 15 Unique Alerts**				
- **Most frequent 5 Unique Alerts**				

Added 2 alert(s) to the Alert cache
Queried on : Thu July 28, 2005 12:52:57
Database: snort@localhost (**Schema Version:** 106)
Time Window: [2005-07-25 17:07:52] - [2005-07-28 12:48:05]

Sensors/Total: 1 / 1
Unique Alerts: 8
Categories: 3
Total Number of Alerts: 83

- Src IP addrs: 7
- Dest. IP addrs: 28
- Unique IP links 33

- Source Ports: 7
  - TCP (7) UDP (0)
- Dest Ports: 2
  - TCP (2) UDP (0)

Traffic Profile by Protocol

TCP (8%)

UDP (0%)

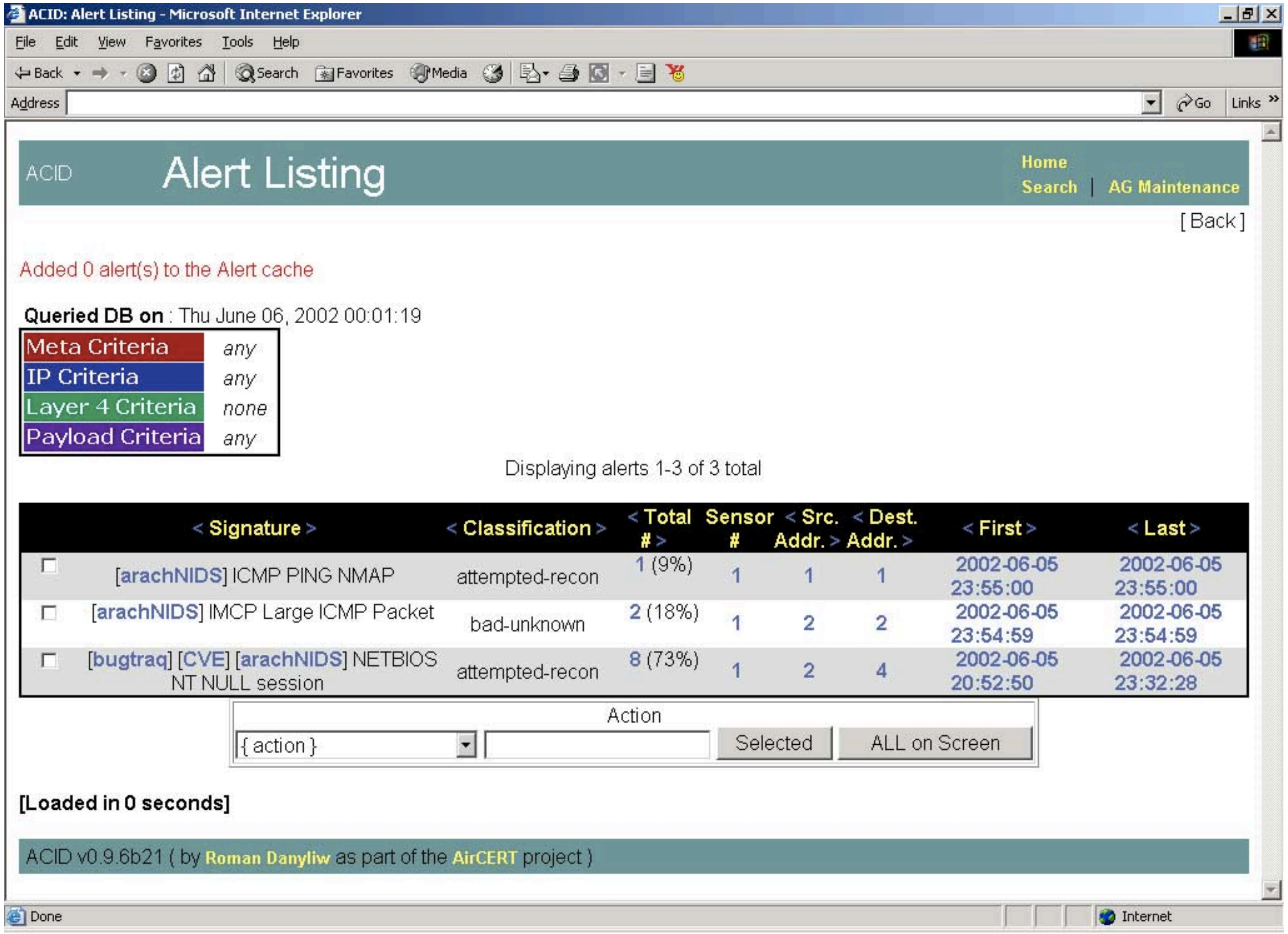
ICMP (31%)

Portscan Traffic (60%)

BASE 1.1.3 (lynn) (by **Kevin Johnson** and the **BASE Project Team**
Built on ACID by Roman Danyliw)

[Loaded in 0 seconds]

BASE (Basic Analysis and Security Engine)



ACID **Alert Listing**

Home
Search | **AG Maintenance**

[Back]

Added 0 alert(s) to the Alert cache

Queried DB on : Thu June 06, 2002 00:01:19

Meta Criteria	*any*
IP Criteria	*any*
Layer 4 Criteria	*none*
Payload Criteria	*any*

Displaying alerts 1-3 of 3 total

	< Signature >	< Classification >	< Total # >	Sensor #	< Src. Addr. >	< Dest. Addr. >	< First >	< Last >
□	[arachNIDS] ICMP PING NMAP	attempted-recon	1 (9%)	1	1	1	2002-06-05 23:55:00	2002-06-05 23:55:00
□	[arachNIDS] IMCP Large ICMP Packet	bad-unknown	2 (18%)	1	2	2	2002-06-05 23:54:59	2002-06-05 23:54:59
□	[bugtraq] [CVE] [arachNIDS] NETBIOS NT NULL session	attempted-recon	8 (73%)	1	2	4	2002-06-05 20:52:50	2002-06-05 23:32:28

Action

{ action } ▾ | Selected | ALL on Screen

[Loaded in 0 seconds]

ACID v0.9.6b21 (by **Roman Danyliw** as part of the **AirCERT** project)

References and documentation

- Snort documentation
    - –http://www.snort.org/docs
- Writing SNORT Rules
    - –http://manual.snort.org/node27.html

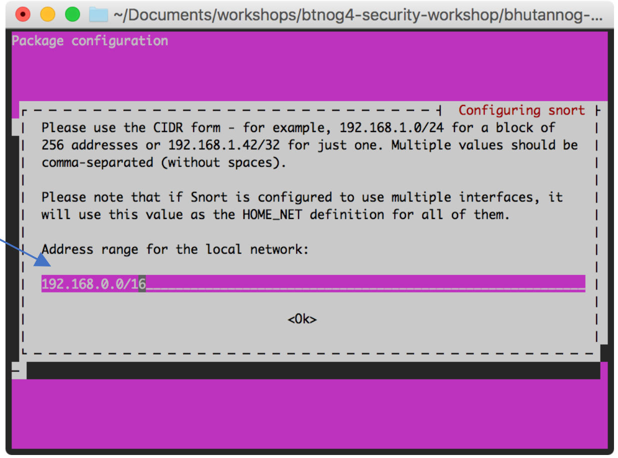
Exercise

SNORT Setup

- ssh to workshop@10.0.0.x
  - Note: x is your group#
  - Note: password is iij/2497

- $ sudo apt install snort
  - Answer Yes to install

- It will ask you for your HOME_NET. For this lab define it as your host IP.
  - 10.0.0.x/32
  - Note x is your group#


~/Documents/workshops/btnog4-security-workshop/bhutannog-...

```
Package configuration

                                                  ┤ Configuring snort ├
 Please use the CIDR form - for example, 192.168.1.0/24 for a block of
 256 addresses or 192.168.1.42/32 for just one. Multiple values should be
 comma-separated (without spaces).

 Please note that if Snort is configured to use multiple interfaces, it
 will use this value as the HOME_NET definition for all of them.

 Address range for the local network:

 192.168.0.0/16

                                  <Ok>
```

SNORT Setup2

- important configuration files
  - SNORT config: /etc/snort/snort.conf
  - SNORT debian config: /etc/snort/snort.debian.conf
  - SNORT rules: /etc/snort/rules

- Check HOME_NET and Interface related configuration from /etc/snort/snort.debian.conf

SNORT Setup3

- $ sudo vi /etc/snort/snort.conf
  - To enable alert log; comment (adding # before the line) the following line:

    output unified2: filename snort.log, limit 128, nostamp, mpls_event_types, vlan_ev ent_types
  - For lab purpose we will disable all predefined rules.
  - Disable (put #) all the line having “include $RULE_PATH” (in Step 7 of configuration file) except ”Include $RULE_PATH/local.rules”
  - Press Esc and type :wq then <enter> to overwrite the file and quit vi

Exercise : 1 - stupid rule

- Adding the following rule into /etc/snort/rules/local.rules

```
alert ip any any -> any any (msg: "IP Packet detected"; sid: 10000;)
```

- $ sudo systemctl restart snort
  - Note: you need to restart snort after changing rules
- Check alerts
- $ sudo tail -f /var/log/snort/snort.alert

- Now delete the rule, and restart snort

Exercise : 2

- Write a rules to check XMAS scan on your server
  - Clue XMAS scan sets the FIN, PSH, and URG flags
  - Check the rules with nmap
    - `nmap -sX SERVER_IP`

Exercise : 3

- Write a rules to check any external network access your webserver /admin pages
  - Match content

Exercise : 4

- Write a rules to check SSH brute force attack and log IP trying to connect more than 3 times in 60 seconds.
  - threshold:type threshold, track by_src, count 3, seconds 60;