Snort - A network intrusion prevention and detection system

What is Snort?

- Snort is an *open source* network intrusion prevention and detection system.
- It uses a rule-based language combining signature, protocol and anomaly inspection methods

- Powerful
  - Stand-alone real-time traffic analysis
  - Packet logging on IP networks
  - Detect a variety of attacks and probes
  - Protocol analysis, content searching/matching
  - Log to a nicely organized, human-readable directory structure
- Flexible
  - Rules language to describe traffic
  - Detection engine utilizes a modular plug-in architecture

Snort Working Modes

1. A packet sniffer: capture and display packets from the network with different levels of detail on the console
2. Packet logger: log data in text file
3. NIDS: network intrusion detection system

Operational modes are configured via command line switches

- Snort automatically tries to go into NIDS mode if no command line switches are given

Using Snort – Sniffer Mode

- Works much like tcpdump
- Decodes packets and dumps them to stdout
- BPF filtering interface available to shape displayed network traffic

What Do The Packet Dumps Look Like?

```
•   =+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

•   11/09-11:12:02.954779 10.1.1.6:1032 -> 10.1.1.8:23
•   TCP TTL:128 TOS:0x0 ID:31237 IpLen:20 DgmLen:59 DF
•   ***AP*** Seq: 0x16B6DA  Ack: 0x1AF156C2  Win: 0x2217  TcpLen: 20
•   FF FC 23 FF FC 27 FF FC 24 FF FA 18 00 41 4E 53  ..#..'..$....ANS
•   49 FF F0                                         I..

•   =+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

•   11/09-11:12:02.956582 10.1.1.8:23 -> 10.1.1.6:1032
•   TCP TTL:255 TOS:0x0 ID:49900 IpLen:20 DgmLen:61 DF
•   ***AP*** Seq: 0x1AF156C2  Ack: 0x16B6ED  Win: 0x2238  TcpLen: 20
•   0D 0A 0D 0A 53 75 6E 4F 53 20 35 2E 37 0D 0A 0D  ....SunOS 5.7...
•   00 0D 0A 0D 00                                   .....

•   =+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Packet Logger Mode

- Multi-mode packet logging options available
  - Flat ASCII, tcpdump, XML, database, etc available
- Log all data and post-process to look for anomalous activity

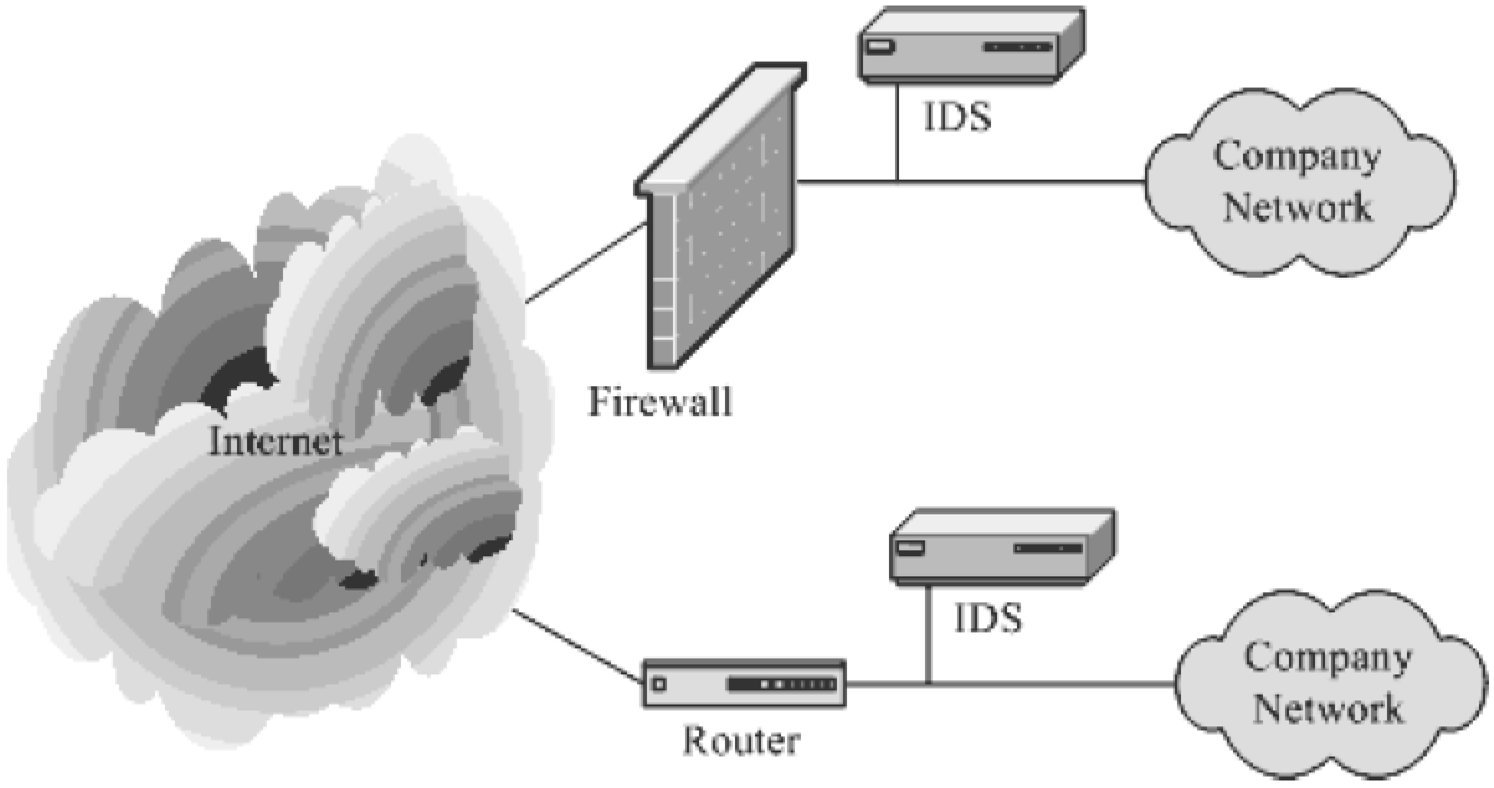
NIDS Mode

- Wide variety of rules available for signature engine
- Multiple detection modes available via rules and plug-ins to analyze traffic for both misuse detection and anomalous activity
- Can perform portscan detection, IP defragmentation, TCP stream reassembly, application layer analysis and normalization, etc

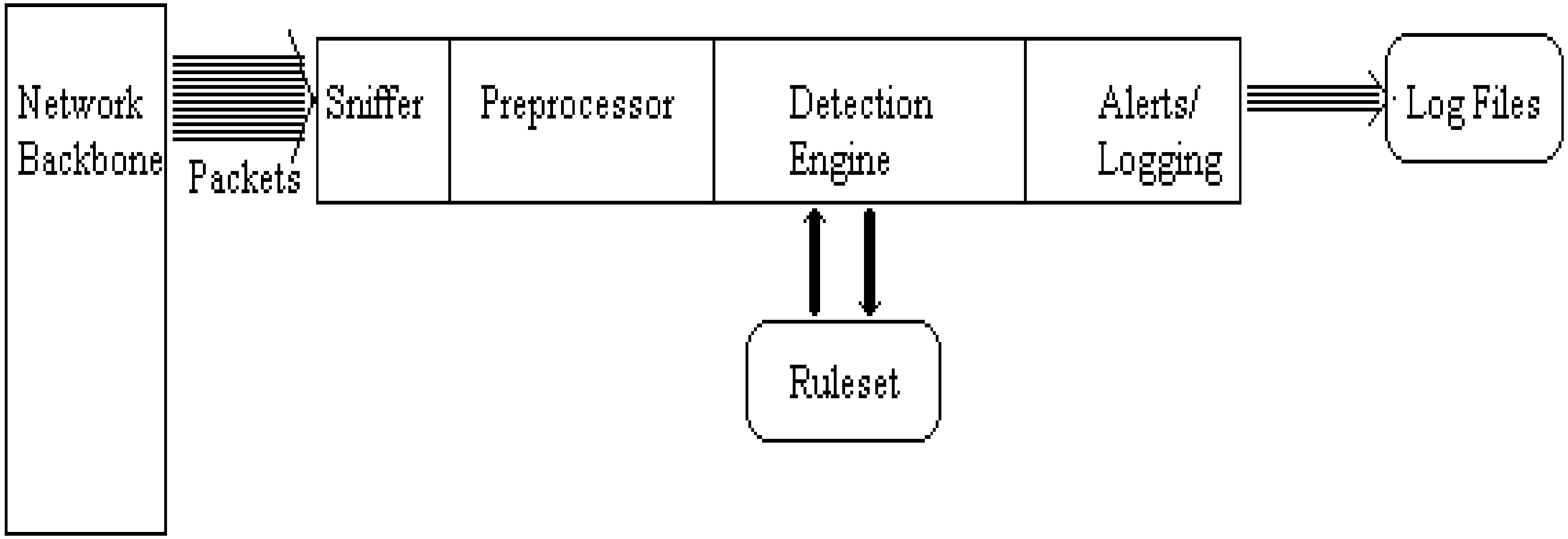
Snort Design

- Packet sniffing network intrusion detection system
- Libpcap-based sniffing interface
- Rules-based detection engine
- Multiple output options
  - decoded logs, tcpdump formatted logs
  - real-time alerting to syslog, file, database, xml

Typical locations for snort



Snort components



- Packet Sniffer
  - Taps into network
  - takes packets from different types of network interfaces , prepare packets for processing

- Preprocessor:
  - Checks against plug-ins
    - RPC plug-in
    - Port scanner plug-in
  - prepare data for detection engine
  - detect anomalies in packet headers
  - packet defragmentation
  - decode HTTP URI
  - reassemble TCP streams.

- Detection Engine:
  - Apply rules to the packets using a string matching algorithm
- The load on the detection engine depends upon the following factors:
  - Number of rules
  - Traffic load on the network
  - Speed of network and machine
  - Efficiency of detection algorithm

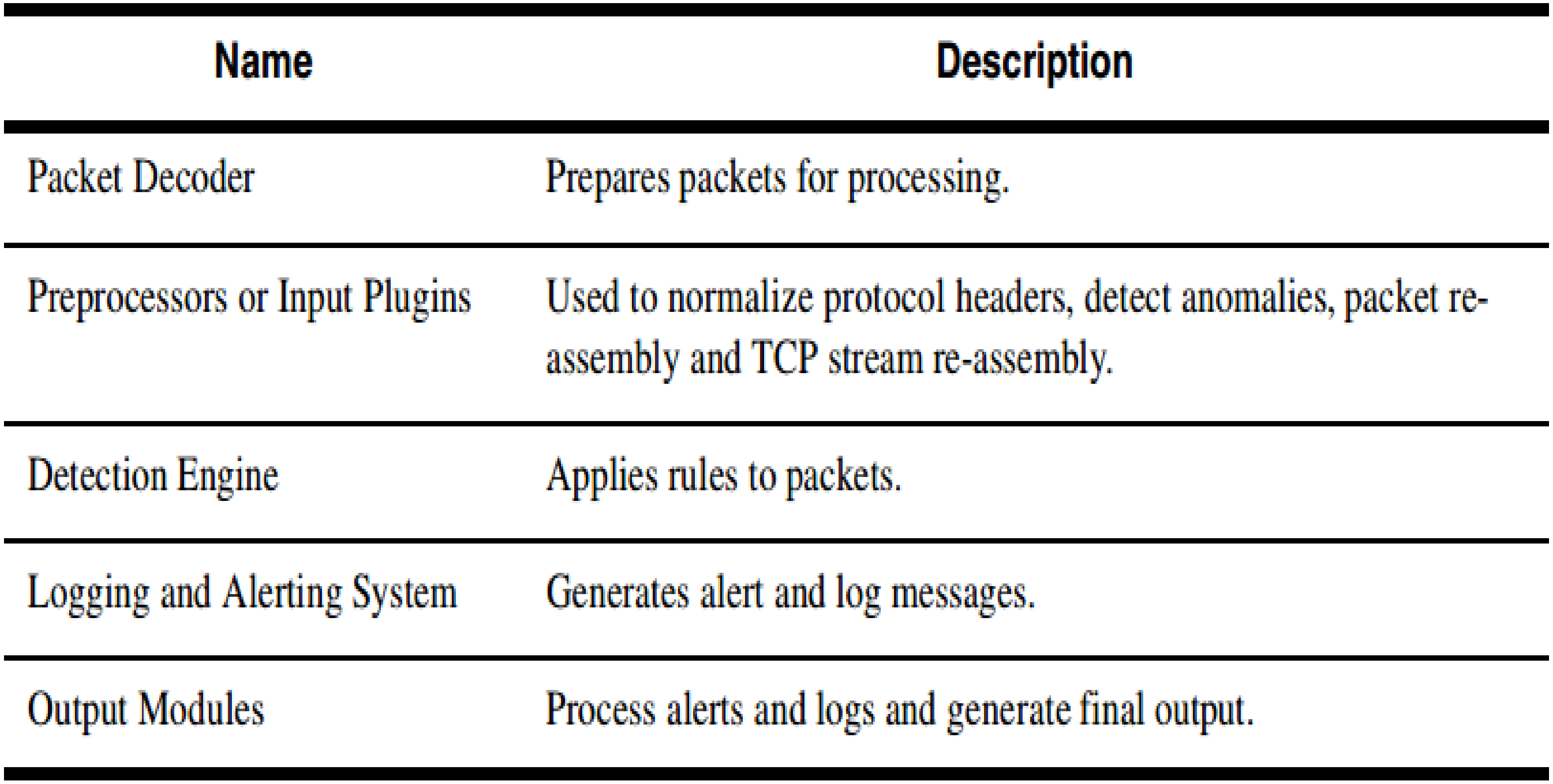
- Snort is a signature-based IDS
  - Implemented via rule-sets
  - Rules
    - Consists of rule header
      - Action to take
      - Type of packet
      - Source, destination IP address

- The detection system can dissect a packet and apply rules on different parts of the packet. These parts may be:
  - The IP header of the packet.
  - The Transport layer header (TCP, UDP, also ICMP)
  - Application layer headers (DNS, FTP, SNMP, and SMTP)
  - Packet payload
    - We can create a rule that is used by the detection engine to find a string inside the data that is present inside the packet.

- Snort Alerting
  - Incoming “interesting packets” are sent to log files.
- Also sent to various Add-ons
  - SnortSnarf (diagnostics with html output)
  - SnortPlot (Perl script that plots attacks)
  - Swatch (provides email alerts).

- **Output Modules**
  Output modules or plug-ins can do different operations depending on how you want to save output generated by the logging and alerting system of Snort.
  - Simply logging to /var/log/snort/alerts file or some other file
  - Sending SNMP traps
  - Sending messages to syslog facility
  - Logging to a database like MySQL or Oracle.
  - Sending Server Message Block (SMB) messages to Microsoft Windows-based machines

Snort: Architecture

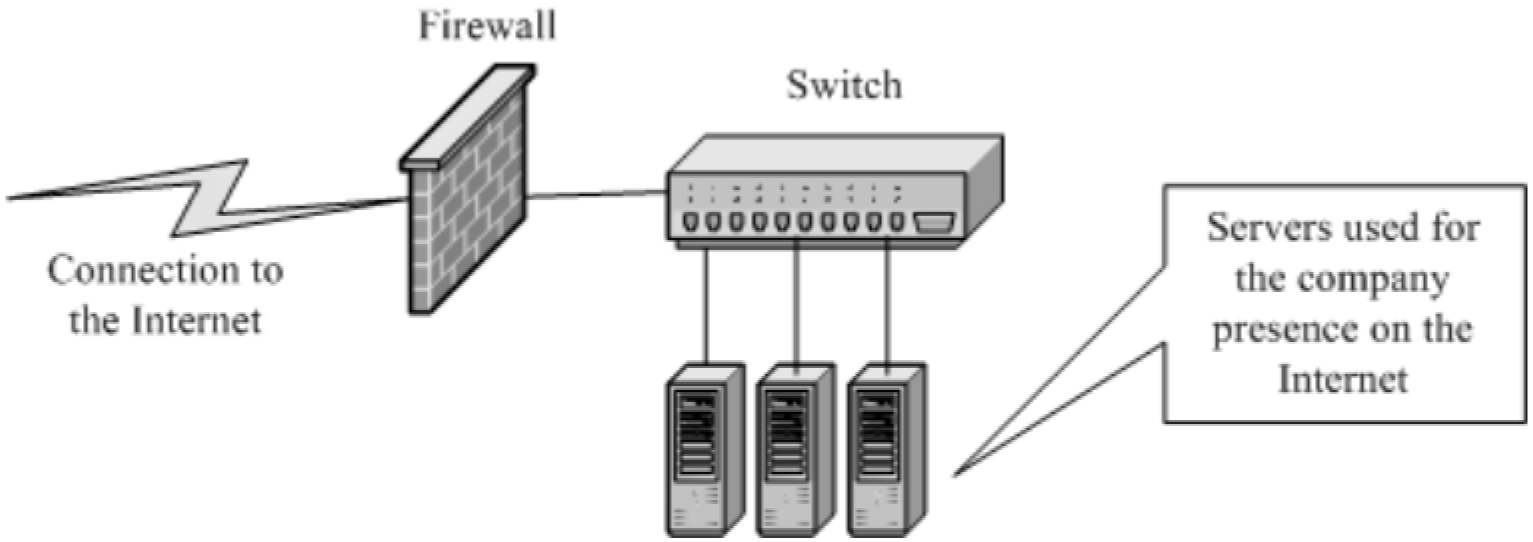


Name	Description
Packet Decoder	Prepares packets for processing.
Preprocessors or Input Plugins	Used to normalize protocol headers, detect anomalies, packet re-assembly and TCP stream re-assembly.
Detection Engine	Applies rules to packets.
Logging and Alerting System	Generates alert and log messages.
Output Modules	Process alerts and logs and generate final output.

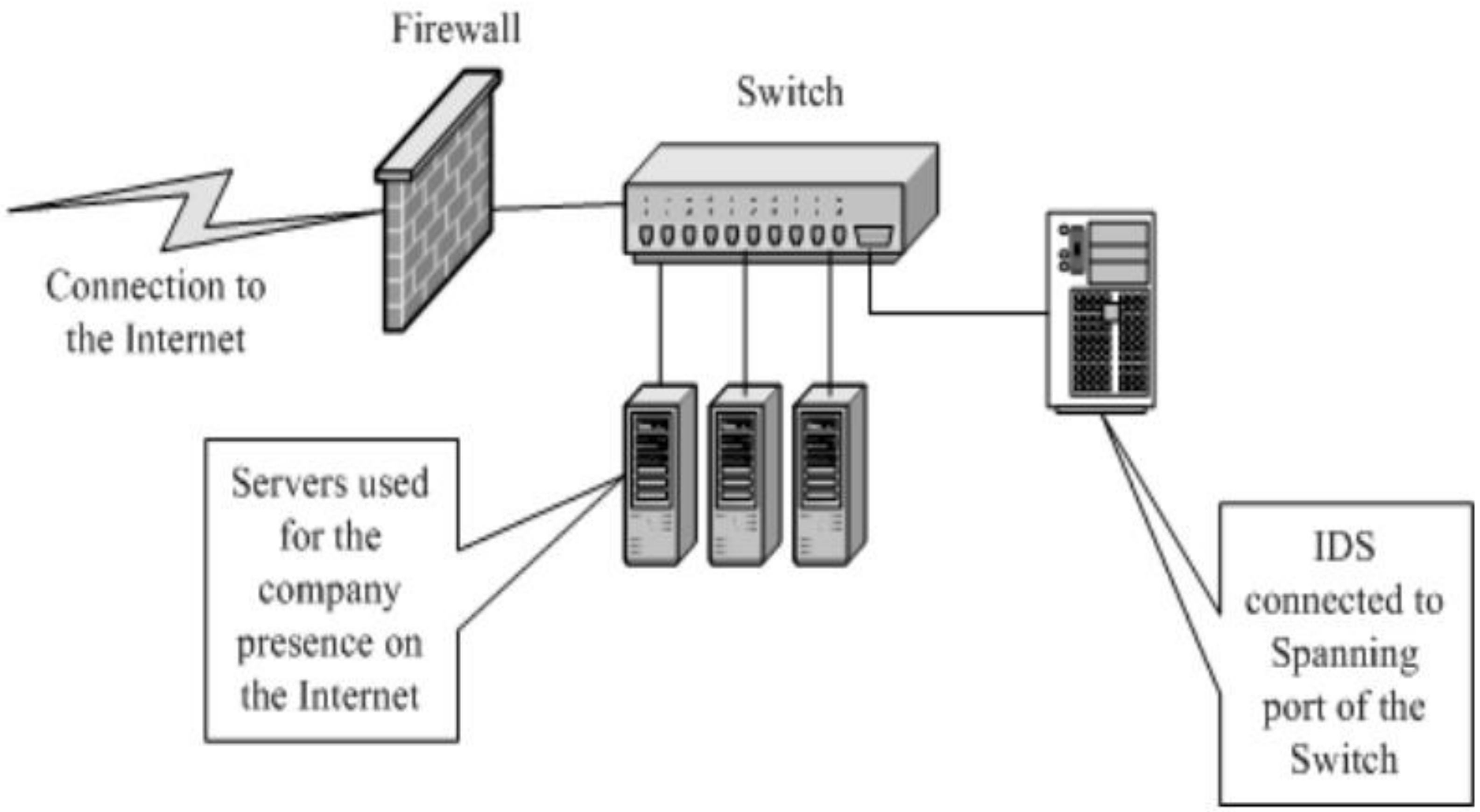
Dealing with Switches

- The best place to install Snort is right behind the firewall or router so that all of the Internet traffic is visible to Snort before it enters any switch or hub.

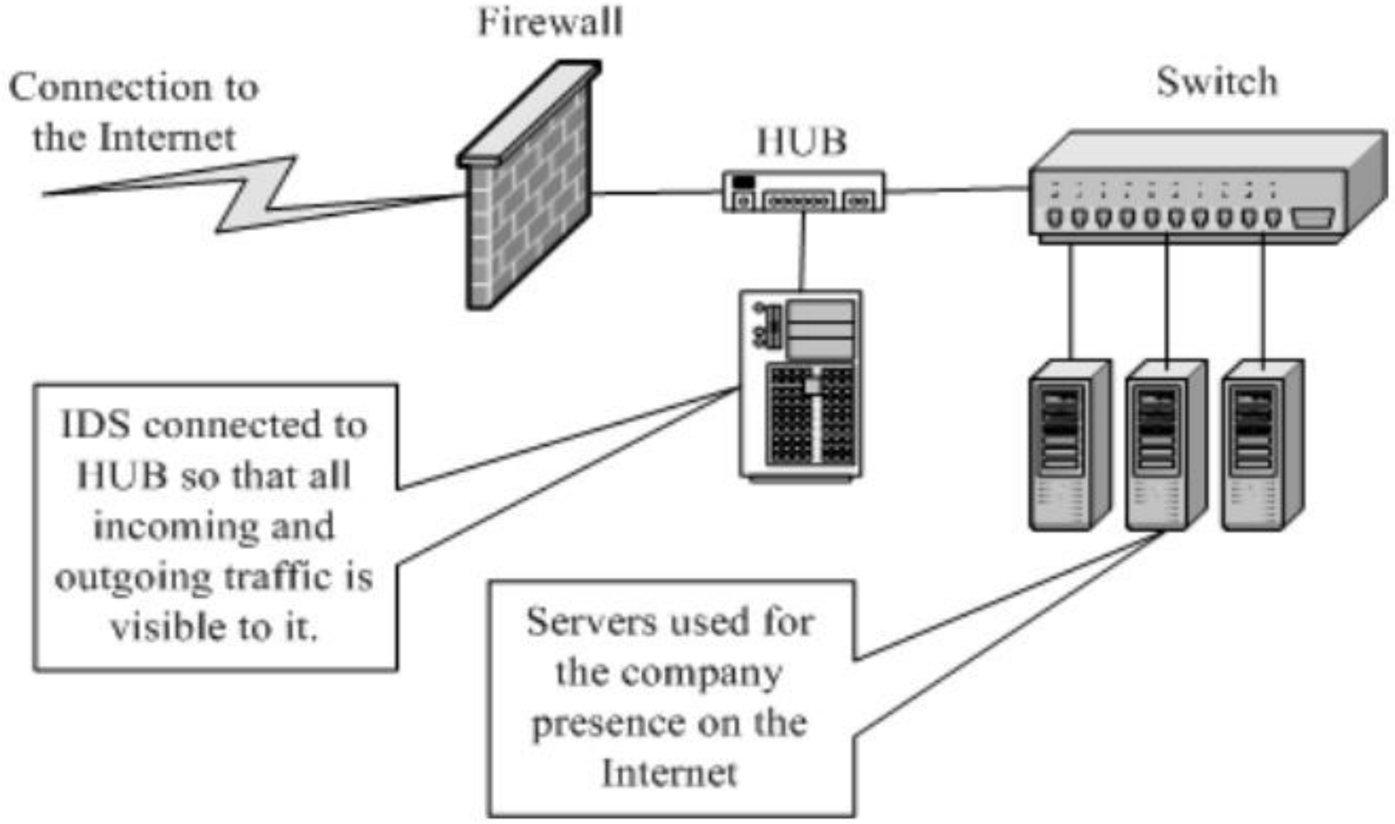
- A typical connection scheme with one firewall and switched network


Firewall
Switch
Connection to the Internet
Servers used for the company presence on the Internet

- The spanning ports allow to replicate all ports traffic on one port where you can attach the Snort machine.
- If the switch has a spanning port, you can connect the IDS machine to the spanning port


Firewall
Switch
Connection to the Internet
Servers used for the company presence on the Internet
IDS connected to Spanning port of the Switch

- You can also connect the IDS to a small HUB or a Network TAP right behind the firewall, i.e., between firewall and the switch.


Firewall
Connection to the Internet
Switch
HUB
IDS connected to HUB so that all incoming and outgoing traffic is visible to it.
Servers used for the company presence on the Internet

Supported Platforms

- Currently Snort is available for the following operating systems:
  - Linux
  - OpenBSD, FreeBSD, NetBSD
  - Solaris (both Sparc and i386)
  - HP-UX
  - AIX
  - IRIX
  - MacOS
  - Windows

How to Protect IDS Itself

- not to run any service on your IDS sensor itself.
- The platform on which you are running IDS should be patched with the latest releases from your vendor.
- Configure the IDS machine so that it does not respond to ping (ICMP Echo type) packets.

- If you are running Snort on a Linux machine, use netfilter/iptable to block any unwanted data.
- You should use IDS only for the purpose of intrusion detection. It should not be used for other activities and user accounts should not be created except those that are absolutely necessary.

Installing Snort

- Snort system utilizes these tools to provide a web-based user interface with a backend database.
  - MySQL is used with Snort to log alert data
  - Apache acts as a web server.
  - PHP is used as an interface between the web server and MySQL database.
  - ACID is a PHP package that is used to view and analyze Snort data using a web browser.
  - GD library is used by ACID to create graphs.
  - PHPLOT is used to present data in graphic format on the web pages used in ACID.

Snort Installation Scenarios

- Depending on the type of network, they can be
  - Test Installation
  - Single Sensor Production IDS
  - Single Sensor with Network Management System Integration
  - Single Sensor with Database and Web Interface
  - Multiple Snort Sensors with Centralized Database

Test Installation

- Consists of a single Snort sensor.
- Snort logs data to text files which viewed later on.
- This arrangement is suitable only for test environments because the cost of data analysis is very high in the production environment.

Single Sensor Production IDS

- A production installation of Snort with only one sensor is suitable for small networks with only one Internet connection.
- You can put sensor behind a router or firewall or put the sensor outside the firewall
- In a production installation, you also need to implement startup and shutdown procedures so that Snort automatically starts at boot time.

Single Sensor with Network Management System Integration

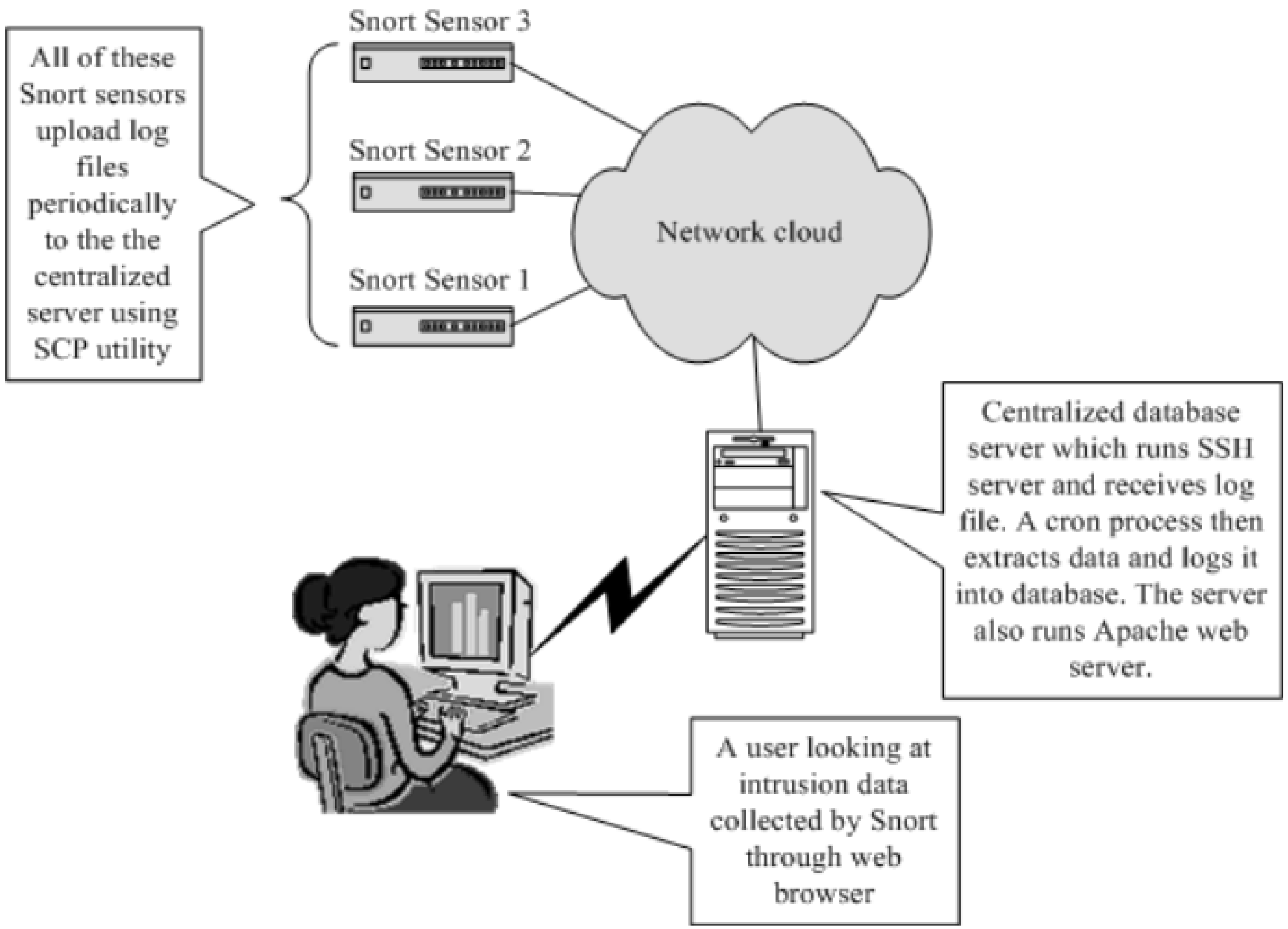
- There are a variety of network management systems used in the enterprise (HP, IBM).
- In a production system, you can configure Snort to send traps to a network management system.
- Snort integration into these network management systems is done through the use of SNMP traps.

Single Sensor with Database and Web Interface

- The most common use of Snort should be with integration to a database.
- The database is used to log Snort data where it can be viewed and analyzed later on, using a web-based interface.
- A typical setup of this type consists of three basic components:
  - Snort sensor
  - A database server
  - A web server

- This setup provides a very good and comprehensive IDS which is easy to manage and user friendly
- You have to provide a user name, password, database name and database server address to Snort to enable it to log to the database.

Multiple Snort Sensors with Centralized Database



All of these Snort sensors upload log files periodically to the the centralized server using SCP utility
Snort Sensor 3
Snort Sensor 2
Snort Sensor 1
Network cloud
Centralized database server which runs SSH server and receives log file. A cron process then extracts data and logs it into database. The server also runs Apache web server.
A user looking at intrusion data collected by Snort through web browser

- Snort sensors do not have a direct connection to the database server.
- The sensors may be configured to log to local files.
- These files can then be uploaded to a centralized server on a periodic basis using utilities like SCP.

- The SCP utility is a secure file transfer program that uses Secure Shell (SSH) protocol.
- The problem with this approach
  is that the data in the database is not strictly real-time. There is a certain delay which depends upon frequency of uploading data using SCP to the centralized database server.

Installing Snort

- Installation of the pre-compiled RPM package
- Installing Snort from Source Code

Installing Snort from the RPM Package

- Involves the following steps
  - **Download**
    Download the latest version from Snort web site (http://www.snort.org).
  - **Install**
    Run the following command to install Snort binaries
    rpm --install snort-1.9.0-1snort.i386.rpm

- This command will perform the following actions:
  - Create a directory /etc/snort where all Snort rule files and configuration files are stored
  - Create a directory /var/log/snort where Snort log files will be stored.
  - Create a directory /usr/share/doc/snort-1.9.0 and store Snort documentation files in that directory.
  - Create a file snort-plain in /usr/sbin directory. This is the Snort daemon.
  - Create a file /etc/rc.d/init.d/snortd file which is startup and shutdown script.

- **Starting, Stopping and Restarting Snort**
  - To run Snort manually, use the following command:
    /etc/init.d/snortd start
  - To stop Snort, use the following command:
    /etc/init.d/snortd stop
  - To restart Snort, use this command:
    /etc/init.d/snortd restart

Installing Snort from Source Code

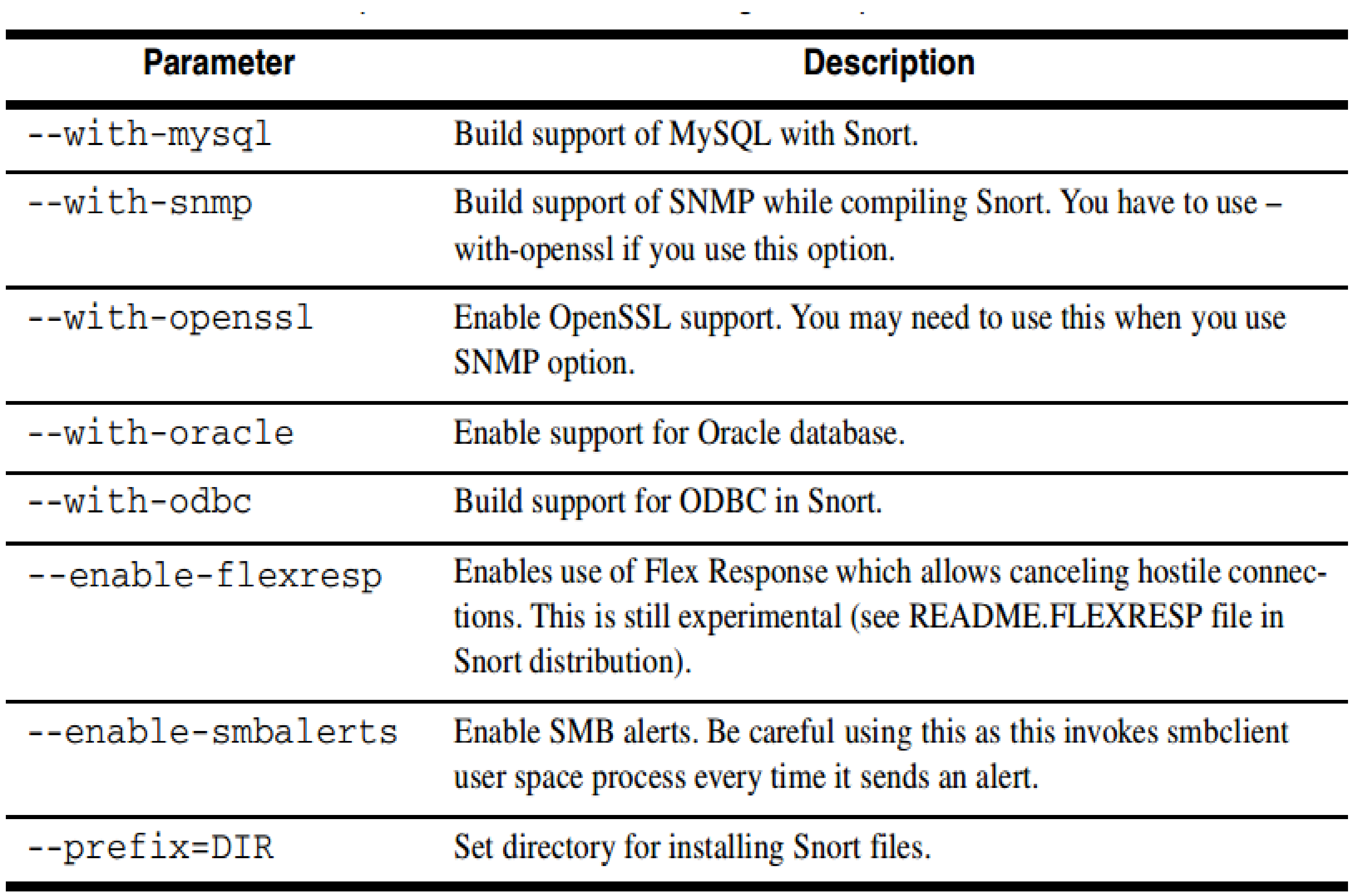
- **Download**
  Download the latest version from Snort web site (http://www.snort.org). The downloadable file name is snort-1.9.0.tar.gz, which can be saved in the /opt directory on the Linux box.
- **Unpacking**
  command to unpack
  ```
  tar zxvf snort-1.9.0.tar.gz
  ```
  This will create a directory /opt/snort-1.9.0

- The directories present under /opt/snort-1.9.0 directory are:
  - The contrib directory contains utilities which are not strictly part of Snort itself. These utilities include ACID, MySQL database creation scripts
  - The doc directory contains documentation files
  - The etc directory contains configuration files.
  - The rules directory contains predefined rule files.
  - All source code is present under the src directory.
  - The templates directory is useful only for people who want to write their own plug-ins.

Compiling and Installation

- Consists of three steps
  - Running the configure script
  - Running the make command.
  - Running the make install command

Command line parameters used with configure scripts

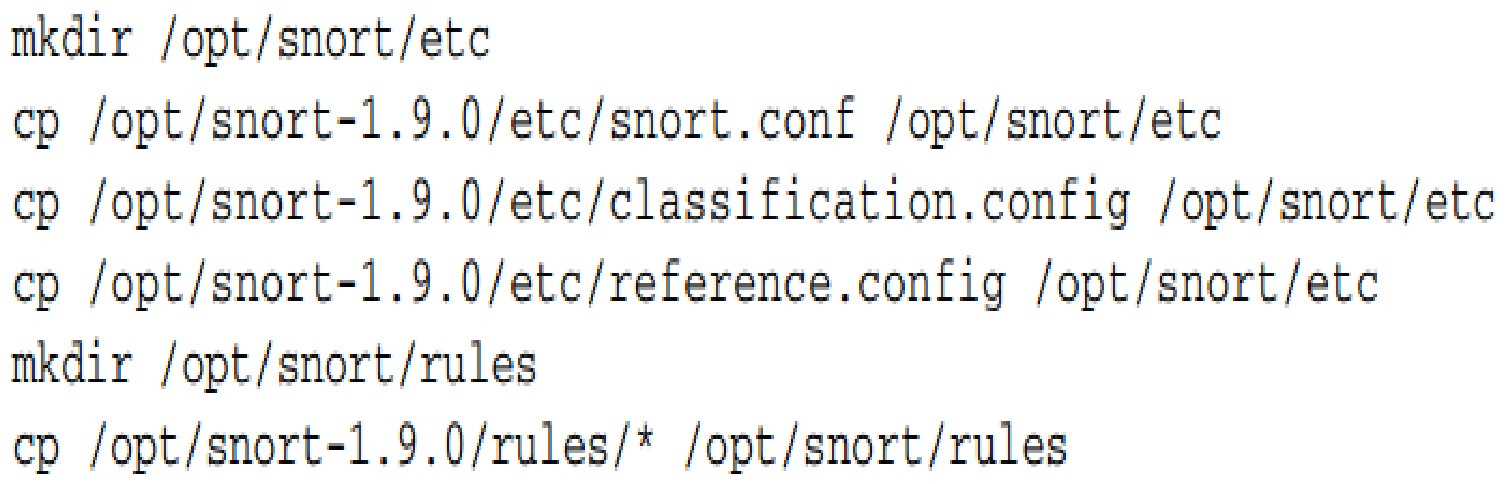


Parameter	Description
`--with-mysql`	Build support of MySQL with Snort.
`--with-snmp`	Build support of SNMP while compiling Snort. You have to use –with-openssl if you use this option.
`--with-openssl`	Enable OpenSSL support. You may need to use this when you use SNMP option.
`--with-oracle`	Enable support for Oracle database.
`--with-odbc`	Build support for ODBC in Snort.
`--enable-flexresp`	Enables use of Flex Response which allows canceling hostile connections. This is still experimental (see README.FLEXRESP file in Snort distribution).
`--enable-smbalerts`	Enable SMB alerts. Be careful using this as this invokes smbclient user space process every time it sends an alert.
`--prefix=DIR`	Set directory for installing Snort files.

After Installation Processes

- Few things are to be done before you can start using Snort.
  - Create directory /var/log/snort where Snort creates log files by default.
  - Create a directory to save configuration files
  - Create or copy the Snort configuration file in /opt/snort/etc directory
  - Create a directory /opt/snort/rules and copy default rule files to /opt/snort/etc directory.

- To perform these actions, you can use the following sequence of commands



```
mkdir /opt/snort/etc
cp /opt/snort-1.9.0/etc/snort.conf /opt/snort/etc
cp /opt/snort-1.9.0/etc/classification.config /opt/snort/etc
cp /opt/snort-1.9.0/etc/reference.config /opt/snort/etc
mkdir /opt/snort/rules
cp /opt/snort-1.9.0/rules/* /opt/snort/rules
```

Snort Rules

- All Snort rules have two logical parts: rule *header* and rule *options*.
- The rule header contains information about what action a rule takes, also contains criteria for matching a rule against data packets.
- The options part usually contains an alert message and information about which part of the packet should be used to generate the alert message.

Rule Header	Rule Options

Basic structure of Snort rules.

Action	Protocol	Address	Port	Direction	Address	Port

Structure of Snort rule header

- The *action* part of the rule determines the type of action taken when criteria are met and a rule is exactly matched against a data packet
- The *protocol* part is used to apply the rule on packets for a particular protocol only.
- The *address* parts define source and destination addresses.

- Source and destination addresses are determined based on direction field.
- In case of TCP or UDP protocol, the *port* parts determine the source and destination ports of a packet on which the rule is applied.

- **Rule Actions**
  - It shows what action will be taken when rule conditions are met.
  - An action is taken only when all of the conditions mentioned in a rule are true.
  - There are five predefined actions.
  - you can also define your own actions as needed.

Predefined actions

- **Pass**
  - This action tells Snort to ignore the packet.
- **Log**
  - The log action is used to log a packet.
- **Alert**
  - The alert action is used to send an alert message when rule conditions are true for a particular packet.
- **Activate**
  - The activate action is used to create an alert and then to activate another rule for checking more conditions.
- **Dynamic**
  - Dynamic action rules are invoked by other rules using the “activate” action

- **User Defined Actions**
  - Sending messages to syslog
  - Sending SNMP traps to a network management system
  - Logging data to XML files
  - Taking multiple actions on a packet

Example

- alert icmp any any -> any any (msg: "Ping with TTL=100"; \ ttl: 100;)
  This is a rule that generates an alert message whenever it detects an ICMP ping packet (ICMP ECHO REQUEST) with TTL equal to 100

- In this rule the action is “alert”, which means that an alert will be generated when conditions are met.
- The protocol is ICMP, which means that the rule will be applied only on ICMP-type packets.
-

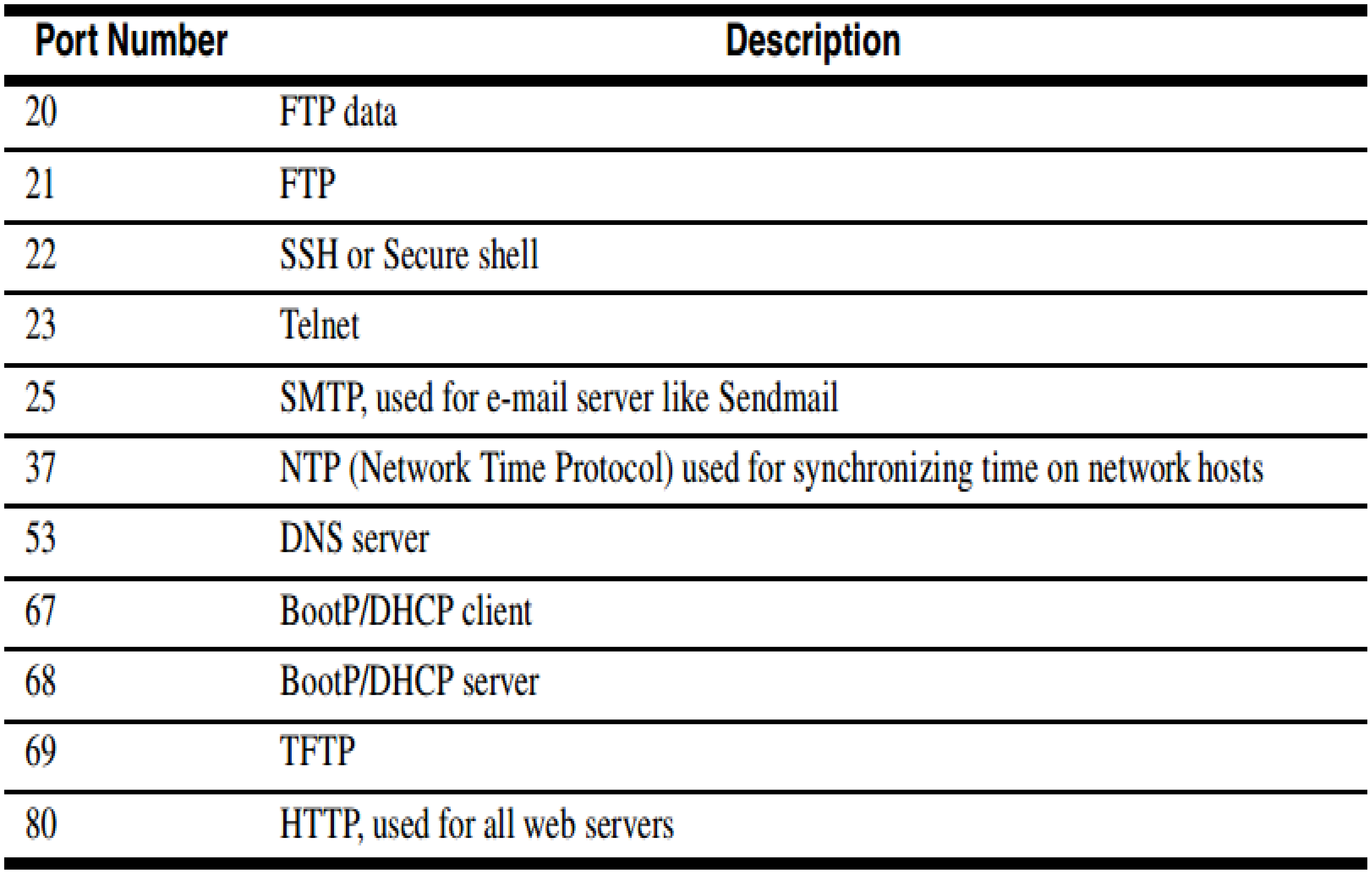
- Source address and source port. In this example both of them are set to any, which means that the rule will be applied on all packets coming from any source.
- direction is set from left to right using the -> symbol
- Destination address and port address. In this example both are set to any, meaning the rule will be applied to all packets irrespective of their destination address

Example

- if you want to generate alerts for all TCP packets with TTL=100 going to web server 192.168.1.10 at port 80 from any source, you can use the following rule:

alert tcp any any -> 192.168.1.10/32 80 (msg: "TTL=100"; \ttl: 100;)

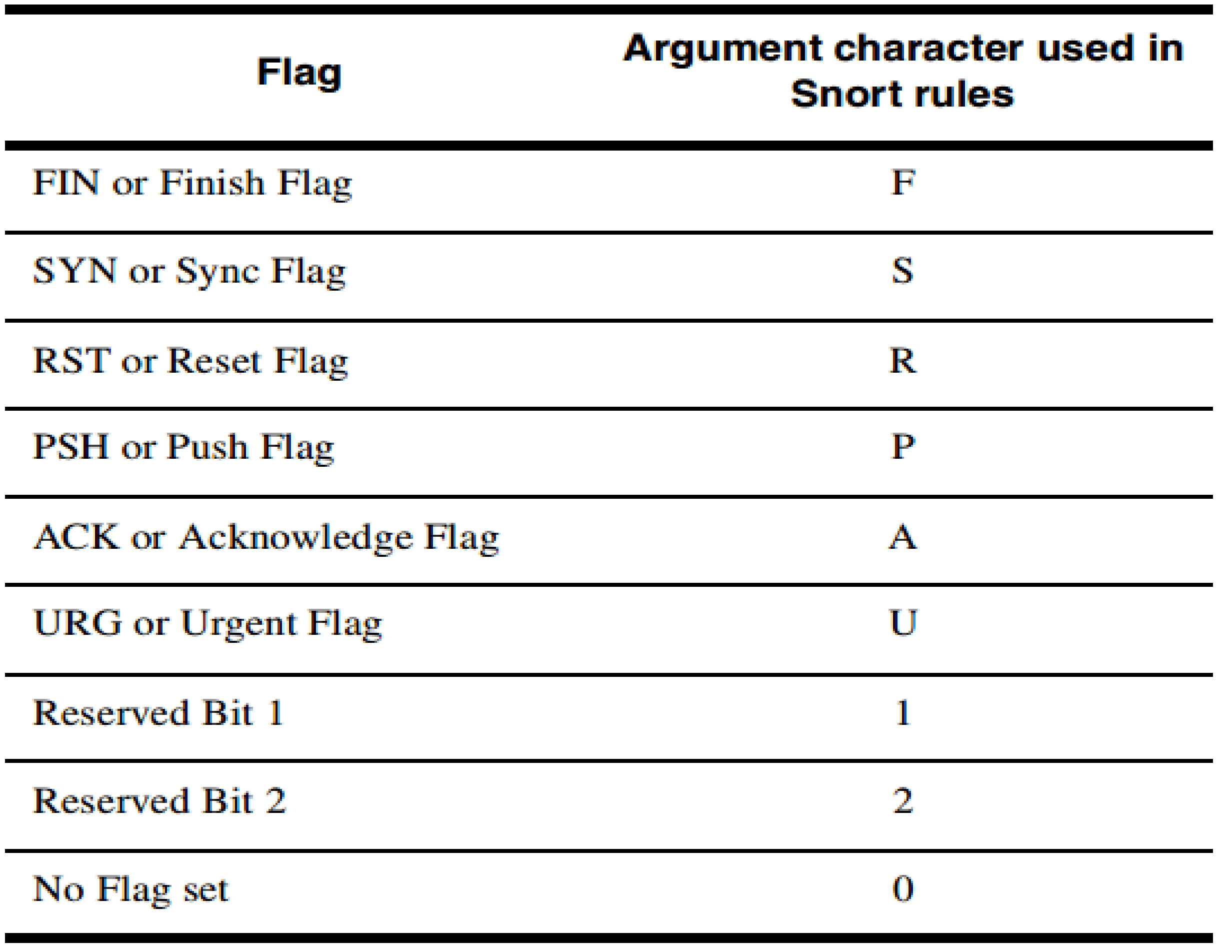
Well-Known Port Numbers



Port Number	Description
20	FTP data
21	FTP
22	SSH or Secure shell
23	Telnet
25	SMTP, used for e-mail server like Sendmail
37	NTP (Network Time Protocol) used for synchronizing time on network hosts
53	DNS server
67	BootP/DHCP client
68	BootP/DHCP server
69	TFTP
80	HTTP, used for all web servers

The flags Keyword

- It is used to find out which flag bits are set inside the TCP header of a packet.



Flag	Argument character used in Snort rules
FIN or Finish Flag	F
SYN or Sync Flag	S
RST or Reset Flag	R
PSH or Push Flag	P
ACK or Acknowledge Flag	A
URG or Urgent Flag	U
Reserved Bit 1	1
Reserved Bit 2	2
No Flag set	0

- Also, ! symbol is used for NOT, + is used for AND, and * is used for OR operation.

Example

- The following rule detects any scan attempt using SYN-FIN TCP packets.
  alert tcp any any -> 192.168.1.0/24 any (flags: SF; \msg: “SYNC-FIN packet detected”;)

- The following rule is used to detect if the DF bit is set in an ICMP packet.
  alert icmp any any -> 192.168.1.0/24 any (fragbits: D; \msg: "Don’t Fragment bit set";)
- The following rule detects if the DF bit is not set
  alert icmp any any -> 192.168.1.0/24 any (fragbits: !D; \msg: "Don’t Fragment bit not set";)

Reference

- R. R. Ur, “Intrusion detection systems with Snort: advanced IDS techniques using Snort, Apache, MySQL, PHP, and ACID”, Prentice Hall Professional, **(2003)**.