

Session 1. Introduction to Information Security

1.1 Introduction

Traditionally we think of Security as protecting the resources under the lock and key .

But in modern world today **Security** is a state of wellbeing (are you secure?)

Security is all about being prepared for the unexpected (challenge) .

And so Information Security is the collection of the following:

1.1.1 Policies

They tell you what to do.

Example: Real life

ID cards

Example: IT world

Password policies

A password must have mixture of numbers, characters, special characters - minimum 8 in length

1.1.2 Procedures

They tell you how to do.

Real life example:

Show the ID card to the guard before entering the premises.

IT world example:

*I. All user **passwords** must contain at least two (2) alphabetic and two-(2) non-alphabetic character. Non-alphabetic characters include numbers (0-9) and special characters.*

II. Procedure to install a package (WordPress), Linux or windows.

1.1.3 Practices

Patterns that are proven to get the desired output.

Real life example

I. Wearing your ID at all times while in premises.

II. Health practices - Washing your hands , covering your face with a mask , changing tooth brush.

IT world example:

Changing Passwords every 2 months, refrain from using easy password like birthdates.

1.1.4 Confidentiality

The state of keeping or being kept secret or private.

Confidentiality prevents the unauthorized use or disclosure of information, ensuring that only those who are authorized to access information can do so.

In other words it is restrictions on the accessibility (who can access the data) and dissemination of the information (if you have the data and you are authorized then you must keep it to yourself).

Real life example - generally , we do not publish our bank account balance on Facebook.

Example in IT world - Keep your password confidential.

1.1.5 Integrity

I. Protecting data from modification or deletion by unauthorized parties.

II. Ensuring that when authorized people make changes , that shouldn't been made, the damage can be undone.

Real life example:

Paying guest --> Room sharing --> Wallet outside --> and no money stolen - Integrity

Example if IT world:

Computer A --> (DATA - ABC) --> Computer B (DATA - ABC) - Integrity maintained

1.1.6 Availability

Ensuring that information is available when required.

Real life example

Your balance (You have money in your BSP account, you can withdraw it when you require cash).

IT World Example:

Sales report saved in the Sales Network folder can be accessed by the sales Managers at all times.

So in a nutshell, INFORMATION SECURITY is the collection of policies, standards, practices to ensure confidentiality, integrity and availability of information.

1.2 Why Information Security

We **need information security** to reduce the risk of unauthorized **information** access, use, disclosure, and disruption.

We **need information security** to reduce risk to a level that is acceptable to the business (management).

We **need information security** to improve the way we do business.

1.3 Security - the money factor involved

1.3.1 How Much Does an Information Security Analyst make?

In developed Countries where security is paramount like USA, Australia, Japan, China,

Information **Security Analysts made** a median salary of \$99,730 in 2019 as per record.

The best-paid 25 percent **made** \$128,640 that year, while the lowest-paid 25 percent **made** \$75,450.

1.3.2 How much Does an Information Security Analyst make in PNG?

Average BSP offer: Base Salary K46 000 + Housing Allowance of K14 000.

Many other PNG companies would offer around the same amount.

1.4 Internet Statistics - Study from a security perspective

Check this out to see the real time internet usage in the world today:

<https://www.internetlivestats.com>

The following link below shows a real-time global *view* of DDoS *attacks*, hacking attempts, and bot assaults mitigated by Imperva security services.

<https://www.imperva.com/cyber-threat-attack-map/>

Session 2. Vulnerability, Threat and Risk

2.1 Vulnerability

It is the weakness in a mechanism that can threaten the wellbeing of an asset.

In other words, it is the lack of Countermeasures of a mechanism or a system.

Real life Example: When you are roaming around without a mask, you are vulnerable to any disease including covid-19.

IT world example: Running a system without a firewall, your system is vulnerable to any attack.

2.1.1 Types of Vulnerabilities

2.1.1.1 Design Vulnerability

When the Vulnerability is said to be inherent to the project or design.

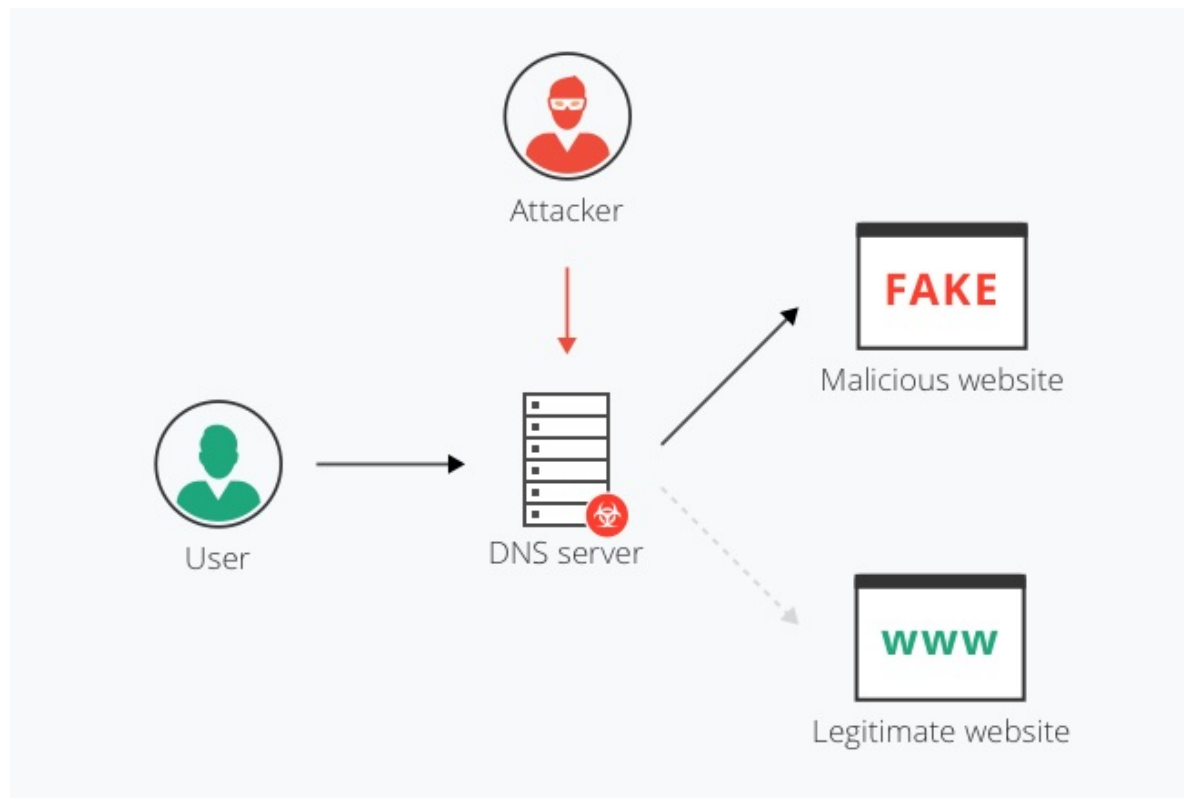
Example -

IP --> overtime IPSec was introduced to address the IP vulnerability.

DNS --> overtime DNSSec was introduced to address the DNS vulnerability.

HTTP --> overtime HTTPS was introduced to address the HTTP vulnerability.

Figure 1 below shows that the Attacker attacks the DNS and redirects the request from the user to a malicious website rather than to a legitimate website. Overtime that design was address with the introduction of the DNSSec (DNS Security Extension).



2.1.1.2 Implementation Vulnerability

Design is good.

Implementation vulnerabilities are also referred to as low-level flaws or technical flaws.

Implementation vulnerabilities encompass several well-publicized vulnerability classes you've probably heard of, such as buffer overflows and SQL injection.

2.1.1.3 Operational / Configurational Vulnerability

A flaw in your security settings, like failing to auto-encrypt your files, could leave your entire network and every device connected to it **vulnerable** to any attack.

One good example of this type of vulnerability is when System Admins fail to reset the default passwords of network devices, e.g. routers. You must change the default provided password of every devices purchased.

2.2 Threat

Threat is any potential danger to your information or system.

Someone uncovering a vulnerability and exploiting it. That person becomes the threat to your organization.

2.3 Risk

Business impact and the probability of a vulnerability being exploited.

For example if you are not carrying a backup regularly in your organization, you are at risk.

In the next section, we will discuss on how we manage risks.

2.3.1 Risk Management

You as the manager for the data center of your organization, one of your prime roles is to (a) identify the risks, (b) assess and prioritize them, (c) control the risks and (d) review the controls.

(A) Identify the Risk

The examples are as follows:

- (1) Virus attack / hacking
- (2) Power failure
- (3) Hardware failure
- (4) Flood / Earthquake
- (5) The system admin - Christmas break to Home province - dies of sorcery/witchcraft

(B) Assess and prioritize the risk (what's important)

(C) Control the risk

- (1) Defensive controls (Firewall, AV, IDS, IPS, Training our staff)
- (2) UPS / Generator
- (3) Use good systems, redundancy, RAID
- (4) DR -Disaster Recovery Location - remotely saved backup to different location
- (5) Junior system - policy - one person at a time

(D) Review the controls (are controls working fine?)

- (1) Maintaining the firewall

Session 3: Exposure, Countermeasures and Firewall

Objective:

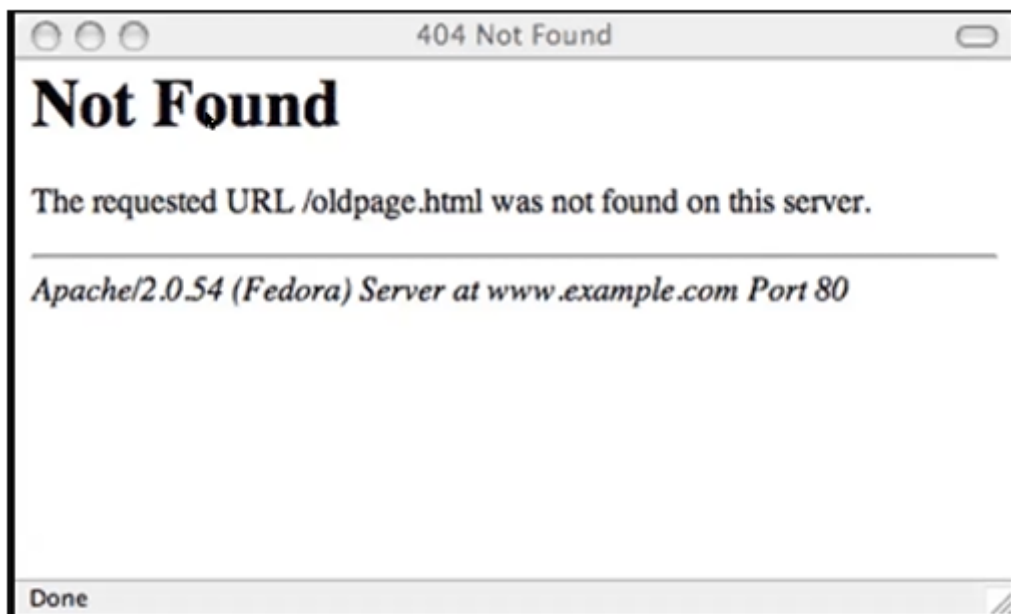
1. Students should be able to understand the term exposure used in security domain.
2. Students should be able to know and apply basic countermeasures to threats.
3. Students should be able to understand the use of firewall in a network.

3.1 Exposure

Exposure is a state of having no protection against something harmful.

Take an example of a BSP user calling out his correct 4-digit bank PIN number at the ATM. This is a very good example of exposure.

Figure 2 below also shows how a certain website's URL and port number and other details are being exposed to public.



3.2 Countermeasures

Countermeasure is the deployment of a set of security to protect against a security threat.

Examples:

- a. Firewall
- b. AV (Anti-Virus)

3.2.1 Firewall

A **firewall** is a **network** security device that monitors incoming and outgoing traffic and decides whether to allow or block specific traffic based on a defined set of security rules. A **firewall** can be hardware, software, or both.

It is a part of an Operating System or Network to :

Block Unauthorized Access,

Implementing policies like blocking social media,

Permit Authorized Access,

Allow only FTP traffic, etc.....

3.2.1.1 Methods of implementing a Firewall

- Permit All (By default)
 - And explicitly denying
- Deny all
 - And explicitly Permit

Session 4: Types of Firewall

4.1 Packet Filtering firewall

The **packet filtering firewall filters** IP **packets** based on source and destination IP address, and source and destination port. The **packet filter** may lack logging facilities, which **would** make it impractical for an organization that has compliance and reporting requirements to which they must adhere.

4.2 Screened Host Firewall

A firewall which is implemented using a firewall router and a proxy server, with the router acting as a front end to the server. The firewall router first screens off any accesses which are disallowed to a closed network, apart from Web page accesses and secure accesses to services such as email. The Web accesses are then passed to the proxy server which acts as a front end to the Web server that actually dispenses the Web pages. There are thus two layers of security that a cracker has to circumnavigate before accessing the Web server inside the closed network.

4.3 Bastion host

A **bastion host** is a special-purpose computer on a network specifically designed and configured to withstand attacks. The computer generally hosts a single application, for example a [proxy server](#), and all other services are removed or limited to reduce the threat to the computer. It is hardened in this manner primarily due to its location and purpose, which is either on the outside of a [firewall](#) or in a demilitarized zone ([DMZ](#)) and usually involves access from untrusted networks or computers.

4.4 Stateful Inspection Firewall

The term **stateful inspection** (also known as the dynamic packet filtering) refers to a **distinguished firewall technology**. It aims to monitor the active connections on a network. Moreover, the process of stateful inspection determines which network packets should be allowed through the firewall by utilizing the information regarding active connections.

4.5 iptables

Iptables is a Linux command line **firewall** that allows system administrators to manage incoming and outgoing traffic via a set of configurable table rules. **Iptables** uses a set of tables which have chains that contain set of built-in or user defined rules.

4.6 Threat Mangement Gateway (TMG)

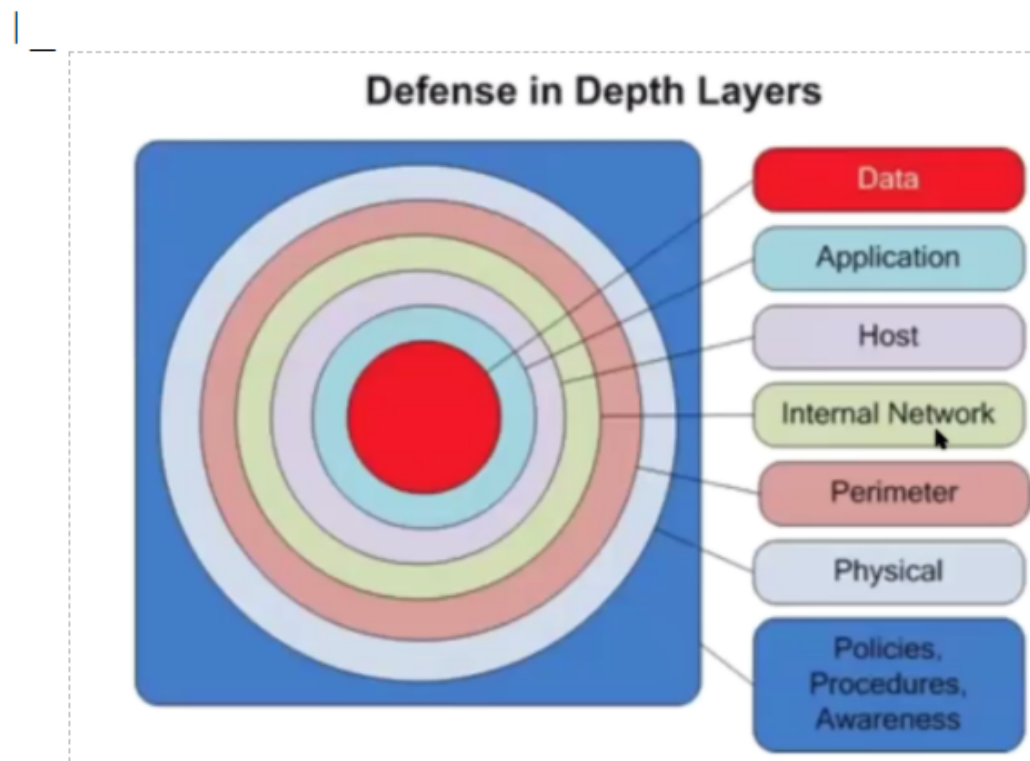
Microsoft Forefront Threat Management Gateway (Forefront **TMG**), formerly known as Microsoft Internet Security and Acceleration Server (ISA Server), is a **network** router, firewall, antivirus program, VPN server and web cache from Microsoft Corporation.

Session 5: Defense in Depth

Also known as Elastic Defense / Castle Approach)

5.1 Defense in depth Layers

Defense in Depth layers



1. Data

|_ Encryption

2. Application

|_ Username and password (ACL)

3. Host

|_ Anti-viruses

Host based Intrusion Detection System (HIDS)

4. Internal Network

|_

Network Intrusion Detection System (NIDS)

5. Perimeter Network

|_

Firewall

6. Physical

|_

Locks and Keys, Bio metric access control Tools

7. Policies, Procedures, Awareness (Infrastructure)

|_ Pre-defined protocols

Like - No Pen drives are allowed in the company

So in a nutshell, the objective of implementing the **Defense in Depth**:

- Each Layer is equipped with a Defense Mechanism
- Attacker has to bypass each layer to get the data
- Delay the attacker to perform the attack, by implementing outer / inner layers .
- If attacks are detected at those layers the inner layers are informed will be ready to take actions or block the attacker .

Lab Segment: Continue lab on iptables:

Adding some more rules:

Task 1 :

Allow incoming connection only from one IP (when INPUT is drop)

```
# iptables -A INPUT -s 192.168.106.127 -j ACCEPT
```

(-s) source IP (cannot be used for ports)

Task 2:

Trust packets from a network (instead of single IP)

```
# iptables -A INPUT -s 192.168.100.0/24 -j ACCEPT
```

or

instead of 24 --> 255.255.255.0

Task 3:

Trust packages from defined iprange

```
# iptables -A INPUT -m iprange --src-range 192.168.10.10-192.168.10.20 -j ACCEPT
```

(iprange) --> Specifying the range

Task 4:

Drop the outgoing packets to specified iprange ,

```
# iptables -A OUTPUT -m iprange --dst-range 192.168.125.100-192.168.125.110 -j DROP
```

Task 5:

Accept the packets from specified IP and MAC,

```
# iptables -A INPUT -s 192.16.120.121 -m mac --mac 00:20:20:20:20:20 -j ACCEPT
```

(-m mac) mac address module

(--mac) mac as an input

NOTE : you cannot use more than one module in a single command

Task 6:

Accept/Allow packets with specific destination port ,

```
# iptables -A INPUT -p tcp --dport 6881 -j ACCEPT
```

or

```
# iptables -A INPUT -p tcp --destination-port 6882 -j ACCEPT
```

(-p) --> protocol

(--dport) --> going to port no

Task 7:

Accept/Allow packets with specific destination port range

```
# iptables -A INPUT -p tcp --dport 6881:6890 -j ACCEPT
```

Task 8:

Accept/Allow packets with specified source port/portrange

```
# iptables -A INPUT -p tcp --sport 6881 -j ACCEPT
```

or

```
# iptables -A INPUT -p tcp --source-port 6882 -j ACCEPT
```

or

```
# iptables -A INPUT -p tcp --sport 6883:6890 -j ACCEPT
```

or

```
# iptables -A INPUT -p tcp --source-port 5883:5890 -j ACCEPT
```

Session 6: Making permanent changes to iptables

SCENE :

Make a permanent change to the iptables so that it won't change if machine is restarted.

(1) write a script --> includes all the lines

```
#!/bin/bash
```

```
iptables -F
```

```
iptables -P INPUT DROP
```

```
iptables -P FORWARD DROP
```

```
iptables -P OUTPUT ACCEPT
```

```
iptables -A INPUT -i lo -j ACCEPT
```

```
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
iptables -L
```

Yes this will also result in restoring the rules/iptables if rebooted, the main motto is to run it every-time when the machine boots up (automated the script to boot)(startup)

(2) Using save and restore command

How to use it

(1) Write a script/step wise commands

```
#!/bi/bash
```

```
iptables -F
```

```
iptables -P FORWARD DROP
```

```
iptables -A INPUT -i lo -j ACCEPT
```

```
iptables -L
```

Run the script

```
# ./script-name
```

(2) Exporting and Importing settings

(1) Create a dir to export/save the settings

```
# mkdir /etc/iptables
```

(2) Export current settings/ip configuration

```
# iptables-save > /etc/iptables/rules.v4
```

(3) Import/restore the saved file

```
# iptables-restore < /etc/iptables/rules.v4
```

or

```
# iptables-restore < /etc/iptables/rules.v6
```

Will still restore the settings if restarted, but restore command saves time to jump to previous seen state.

(3) Using Package iptables-persistent (kali --> netfilter-persistent)

```
# apt-get install iptables-persistent
```

(persistent) --> runs # iptables-restore < /etc/iptables/rules.v4 on the startup

Session 7: Wireshark (Lab session)

Wireshark is the world's foremost and widely-used network protocol analyzer. It lets you see what's happening on your network at a microscopic level and is the standard across many commercial and non-profit enterprises, government agencies, and educational institutions.

Wireshark has a rich feature set which includes the following:

- Deep inspection of hundreds of protocols, with more being added all the time

- Live capture and offline analysis
- Standard three-pane packet browser
- Multi-platform: Runs on Windows, Linux, macOS, Solaris, FreeBSD, NetBSD, and many others
- Captured network data can be browsed via a GUI, or via the TTY-mode TShark utility
- The most powerful display filters in the industry
- Capture files compressed with gzip can be decompressed on the fly
- Coloring rules can be applied to the packet list for quick, intuitive analysis and many more.

So in simple terms, Wireshark is simply a Network Packet Capturing Tool (to analyze the packet) .

It is a Open Source Tool and can be installed on Windows OS, Mac OS or Linux OS.

Tasks:

1. Downloading and Installation of Wireshark
2. Installation steps to be guided by the Trainer.
3. Create a filter for data collection and display
4. Examine real world Packet Capture:
 - a. Ping Windows machine from Linux machine. Capture IP packets (specifically ICMP packets) on Wireshark.
 - b. ssh to Linux machine on Windows machine and capture ssh packets on Wireshark
nb: ssh uses default port of 22

Observation:

There are three windows that are separated

1. Packet List
2. Packet Details
3. Packet Bytes

Session 8: tcpdump - Linux based packet capturing tool

It is the first generation IDS (Intrusion Detection System).

Lab Segment:

1. Is tcpdump already installed?

which tcpdump

If not installed :

```
# apt-get update
# apt-get install tcpdump
# which tcpdump ( Verify )
```

2. Getting Started

a. To start packet capturing just run:

tcpdump

To stop Ctrl + C | Ctrl + Z

b. To display all the available interfaces for tcpdump

```
# tcpdump -D
```

Selecting the interface (If you do not select anything it'll grab the first one listed)

```
# tcpdump -i ens33
```

(-i)--> interface

(ens33) --> interface name

Filters & commands

It only supports a single filter unlike wireshark

```
# tcpdump udp
```

(udp) --> acts as a filter

Mentioning the interface with the filter type

```
# tcpdump -i ens33 udp
```

limiting the no of outputs

```
# tcpdump -i ens33 udp -c 2
```

(-c 2) --> this will limit the output to 2 packets , where c is count

Limiting the time

```
# timeout 2 tcpdump
```

(2) --> 2 sec

Output in strictly numerical output

```
# tcpdump -i ens33 udp -c 4 -n
```

(-n) --> no name resolution (no only)

Writing to a file

tcpdump -i ens33 icmp -c 4 -n -w file1.pcap

(-w) --> used to write output to the file

Reading the file

```
# tcpdump -r file1.pcap
```

(-r) --> reading the file

tcpdump for a specific ip address and port number:

```
tcpdump -i eth0 host 192.168.56.101 and port 5060 -n -s 0 -vvv -w /usr/src/dump
```

NOTE - we can use (> , >>) to write a file but the output format should be ASCII (*also cat command is used to read the ASCII files), but our output is in binary format (the file file1.pcap has binary formats)

Session 9: Snort for Windows (1)

Snort is the foremost Open Source Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) in the world. Snort IDS and IPS uses a series of rules that help define malicious network activity and uses those rules to find packets that match against them and generates alerts for users.

Snort can be deployed inline to stop these packets, as well. Snort has three primary uses: As a packet sniffer like tcpdump, as a packet logger — which is useful for network traffic debugging, or it can be used as a full-blown network intrusion prevention system. Snort can be downloaded and configured for personal and business use alike.

Snort installation on Windows

9.1 Installation of both prerequisites and the snort IDS

(1) To Install snort (Snort_2_9_2_3_Installer.exe)

|_

Snort requires (WinPcap_4_1_3.exe) . Download and install on your machine.

(2) Explore and understand the snort Directory

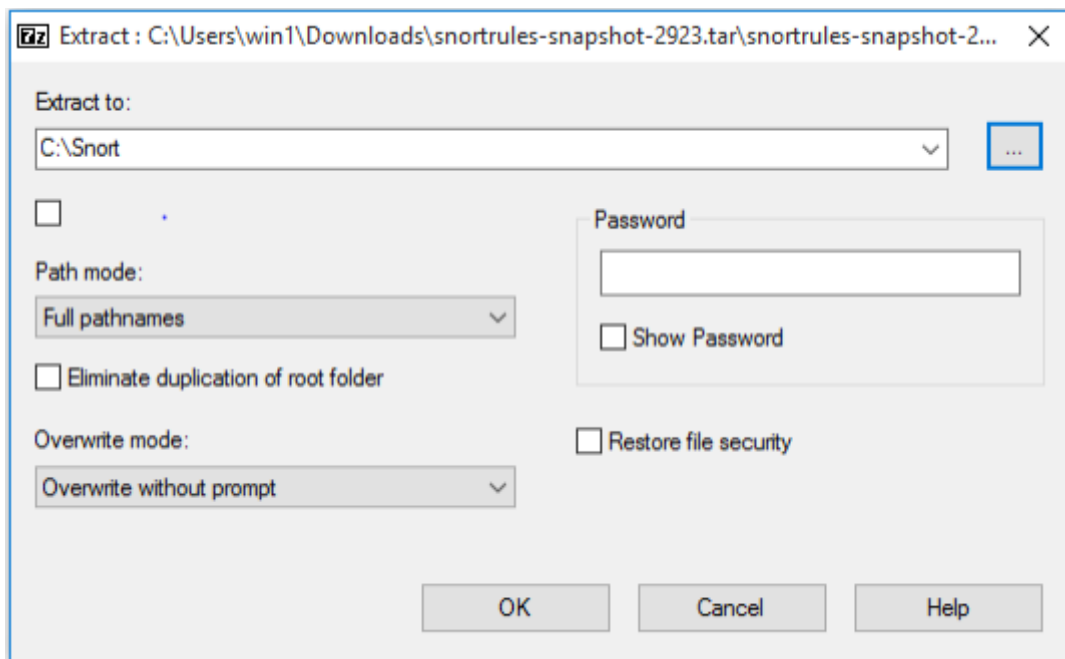
(3) Extracting Rules (Disable antivirus / defender)

Here the rules file is (snortrules-snapshot-2923.tar.gz). Download from your ftp site.

The recommended extraction tool is 7zip. Please download and install it on your machine.

(i) First extract it (gunzip)

(ii) Now extract the .tar as following



Check the rules folder in the snort directory to ensure the rules are successfully placed in there.

(4) Download and install notepad++ on your machine. Notepad++ plays an important role in the next session, that is configuration and Testing of Snort IDS.

Session 10: Configuration and Testing of Snort IDS

(1) Commands (CMD run as admin)

Get inside this Directory (C:\Snort\bin)

snort -W (to show the interfaces/adapters)

Run snort in Packet capturing mode

snort -i 1 (index no of adapter)

CTRL + C to exit

snort -dev (combined)

(-d) TO display the application layer data

(-e) To display the link layer data

(-v) To run in verbose mode (more data)

#snort -i 1 -c c:\snort\etc\snort.conf -l c:\snort\log -A console

(-c) To specifit configuration file to use

(-l) To specify the log location

(-A) To show alert

(console) To show alert to the console

Error 1	ERROR: C:\Snort\etc\Snort.conf(45) Unknown rule type: ipvar.

Open Notepad ++ and open the snort.conf file

replace ipvar --> var (in snort.conf file)

Error 2	ERROR: C:\Users\sssturges\snort-2.9.2.3\src\parser.c(5302) Could not stat dynamic module path "/usr/local/lib/snort_dynamicpreprocessor/": No such file or directory.

Search for the line in Notepad++ (snort_dynamicpreprocessor)

Spoon feed it via giving all the file paths and not the whole directory

NOTE : easy method to copy all the file names is

in CMD go to the file location (C:\Snort\lib\snort_dynamicpreprocessor) and do (# dir /b)

see below:

```
C:\Snort\lib\snort_dynamicpreprocessor>dir/b
sf_dce2.dll
sf_dnp3.dll
sf_dns.dll
sf_ftptelnet.dll
sf_gtp.dll
sf_imap.dll
sf_modbus.dll
sf_pop.dll
sf_reputation.dll
sf_sdf.dll
sf_sip.dll
sf_smtp.dll
sf_ssh.dll
sf_ssl.dll
```

Now add the location lines in front of the file names

```
247 # dynamicpreprocessor directory /usr/local/lib/snort_dynamicpreprocessor/
248
249 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sف_dce2.dll
250 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sف_dnp3.dll
251 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sف_dns.dll
252 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sف_ftptelnet.dll
253 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sف_gtp.dll
254 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sف_imap.dll
255 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sف_modbus.dll
256 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sف_pop.dll
257 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sف_reputation.dll
258 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sف_sdf.dll
259 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sف_sip.dll
260 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sف_smtp.dll
261 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sف_ssh.dll
262 dynamicpreprocessor C:\Snort\lib\snort_dynamicpreprocessor\sف_ssl.dll
263
```

ERROR 3	ERROR: C:\Users\ssturges\snort-2.9.2.3\src\parser.c(5302) Could not stat dynamic module path "/usr/local/lib/snort_dynamicengine/libsf_engine.so": No such file or directory.

In notepad++ search for libsf_engine.so

The location it is looking for is wrong as well as the file name

Change the filename (in notepad ++) libsf_engine.so --> sf_engine.dll

The correct path is (dynamicengine C:\Snort\lib\snort_dynamicengine\sف_engine.dll) (On line 256)

ERROR 4	ERROR: C:\Users\ssturges\snort-2.9.2.3\src\parser.c(5302) Could not stat dynamic module path "/usr/local/lib/snort_dynamicrules": No such file or directory.

The folder it is looking for is not available in the lib folder

Create a folder named (snort_dynamicrules)

In Notepad ++ update the line as (Line 269)

(dynamicdetection directory C:\Snort\lib\snort_dynamicrules)

ERROR 5	ERROR: C:\Snort\etc\Snort.conf(281) Unknown preprocessor: "normalize_ip4".

In Notepad ++ search for this line (normalize_ip4)

280	# Does nothing in IDS mode
281	# preprocessor normalize_ip4
282	# preprocessor normalize_tcp: ips ecn stream
283	# preprocessor normalize_icmp4
284	# preprocessor normalize_ip6
285	# preprocessor normalize_icmp6

Comment the all five lines.

ERROR 6	ERROR: C:\Snort\etc\Snort.conf(527) => Unable to open address file C:\Snort\etc..\rules\white_list.rules, Error: No such file or directory

In the rules folder (C:\Snort\rules), There is no rule as (white_list.rules)

Goto the folder (C:\Snort\rules) and create a file as (white_list.rules)

ERROR 7	ERROR: C:\Snort\etc\Snort.conf(527) => Unable to open address file C:\Snort\etc..\rules\black_list.rules, Error: No such file or directory

In the rules folder (C:\Snort\rules), There is no rule as (black_list.rules)

Go to the folder (C:\Snort\rules) and create a file as (black_list.rules)

Finally run this command again

C:\Snort\bin>snort -i 1 -c C:\Snort\etc\Snort.conf -l C:\Snort\log -A console

If you see PID Allocated, then you did a good configuration. you almost there.

More Editing in snort.conf

Lines 105,106,107

Change

```
var RULE_PATH ../rules
var SO_RULE_PATH ../so_rules
var PREPROC_RULE_PATH ../preproc_rules
```

to

```
var RULE_PATH C:\Snort\rules
var SO_RULE_PATH C:\Snort\so_rules
var PREPROC_RULE_PATH C:\Snort\preproc_rules
```

Line 113, 114

Change

```
var WHITE_LIST_PATH ../rules
var BLACK_LIST_PATH ../rules
```

to

```
var WHITE_LIST_PATH ../rules
var BLACK_LIST_PATH ../rules
```

Lines 524,525 (slash symbol)

Change

```
whitelist $WHITE_LIST_PATH/white_list.rules, \  
blacklist $BLACK_LIST_PATH/black_list.rules
```

to

```
whitelist $WHITE_LIST_PATH\white_list.rules, \  
blacklist $BLACK_LIST_PATH\black_list.rules
```

Line 572

Change

```
include $RULE_PATH/blacklist.rules
```

to

```
include $RULE_PATH/black_list.rules
```

Run it as test conf file for errors (-T)

```
# snort -i 1 -c c:\Snort\etc\snort.conf -l c:\snort\log -T
```

The message you should see if all the configuration is done perfectly is:

```
Snort successfully validated the configuration!  
Snort exiting
```

Session 11: Introduction to network monitoring System- Nagios

Overview

First launched in 1999, Nagios has grown to include thousands of projects developed by the worldwide Nagios community. Nagios is officially sponsored by Nagios Enterprise in a number of different ways through sales of its commercial products and services.

Nagios monitors your entire IT infrastructure to ensure systems, applications, services, and business processes are functioning properly. In the event of a failure, Nagios can alert technical staff of the problem, allowing them to begin remediation processes before outages affect business processes, end-users, or customers. With Nagios you'll never be left having to explain why an unseen infrastructure outage hurt your organization's bottom line.

Capabilities

Nagios provides complete network monitoring capabilities. Nagios monitors the network for problems caused by overloaded and/or crashed servers, network connections or other devices. Easily able to monitor availability, uptime and response time of every node on the network, Nagios can deliver the results in a variety of visual representations and reports.

Benefits

Implementing effective network monitoring with Nagios offers the following benefits:

- Agentless monitoring
- Determine unresponsive machine and alert staff immediately
- Increased server, services, and application availability
- Fast detection of network outages and protocol failures

Session 12: Installation and configuration of Nagios

Nagios (No GUI)

|_ Also known as Nagios core

Developed in 'c' language

1. Make sure you sync the timezone of your server

dpkg-reconfigure tzdata (Asia -> Kol)

2. Install Dependencies

|_

```
# apt-get install apache2 apache2-utils autoconf gcc libc6 libgd-dev make php python  
python3 tree unzip wget
```

3. Download and extract Nagios source code (nagios-4.4.5)

|_

link

|_ Download in from your ftp site.

extract the file

|_

```
#cd Downloads
```

```
#tar -zxf nagios-4.4.5.tar.gz
```

4. Create user and Group

```
#useradd nagios  
#groupadd nagcmd  
#usermod -a -G nagcmd nagios  
#usermod -a -G nagcmd www-data
```

5. Configure the Nagios site

```
#tree -d /etc/apache2/sites-* (this is to check the present site files)
```

```
# ./configure --with-httpd-conf=/etc/apache2/sites-enabled  
# make all  
# make install  
# make install-daemoninit  
sudo make install-config  
sudo make install-commandmode  
sudo make install-webconf
```

6. Apache2 Enable module

```
|_
```

```
# a2enmod rewrite
```

```
# a2enmod cgi
```

7. Create a user for protecting the website (nagiosadmin)

```
|_ # htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

8. Testing the nagios configuration file

```
|_ # /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

(-v) is to verify

9. Start apache

```
|_ # systemctl restart apache2
```

10. Start nagios

```
|_ # systemctl start nagios.service
```

11. On the browser (same machine)

```
|_ 192.168.106.150/nagios or localhost/nagios
```

Enter the server ip

- It'll ask for password
- Enter the password you have created during configuration.
- Username is : nagiosadmin and password is : nagios@123 or your own favorite password.

Session 13: Installation of Nagios Plugins

Steps of Installation of Nagios Plugins

|_

1. Install the pre-requisite

```
# apt-get install automake autotools-dev bc build-essential dc gawk gettext libmcrypto-dev libnet-snmpp-perl libssl-dev snmp
```

Download and Extract Nagios Plugin Source code

|_

extract the file

change the dir

compile and install

```
ls -l /usr/local/nagios/libexec
```

Run:

```
./tools/setup
```

configure script is available now

from extracted folder

```
./configure
```

```
make
```

```
make install
```

```
ls /usr/local/nagios/libexec (should contain multiple files)
```

```
systemctl restart apache2
```