Dr. D. Y. Patil Pratishthan's

Institute for Advanced Computing &Software Development

IACSD

Network Defense and Countermeasures

INDEX

Security

Cyberspace (internet, work environment, intranet) is becoming a dangerous place for all organizations and individuals to protect their sensitive data or reputation. This is because of the numerous people and machines accessing it. It is important to mention that the recent studies have shown a big danger is coming from internal threats or from disappointed employees, another internal threat is that information material can be easy accessible over the intranet.

One important indicator is the IT skills of a person that wants to hack or to breach your security has decreased but the success rate of it has increased, this is because of three main factors −

- Hacking tools that can be found very easily by everyone just by googling and they are endless.
- Technology with the end-users has increased rapidly within these years, like internet bandwidth and computer processing speeds.
- Access to hacking information manuals.

All this can make even a school boy with the curiosity, a potential hacker for your organization.

Since locking down all networks is not an available option, the only response the security managers can give is to harden their networks, applications and operating systems to a reasonable level of safety, and conducting a business disaster recovery plan.

What to Secure?

You are an IT administrator in a small company having two small servers staying in a corner and you are very good at your job. You are doing updates regularly, setting up firewalls, antiviruses, etc. One day, you see that the organization employees are not accessing the systems anymore. When you go and check, you see the cleaning lady doing her job and by mistake, she had removed the power cable and unplugged the server.

What I mean by this case is that even physical security is important in computer security, as most of us think it is the last thing to take care of.

- Data Encryption
- Virus Protection
- Secure Data Exchange
- Data Storage

Now let's go directly to the point of what all to secure in a computer environment −

- First of all, is to check the physical security by setting control systems like motion alarms, door accessing systems, humidity sensors, temperature sensors. All these components decrease the possibility of a computer to be stolen or damaged by humans and environment itself.
- People having access to computer systems should have their own user id with password protection.
- Monitors should be screen saver protected to hide the information from being displayed when the user is away or inactive.
- Secure your network especially wireless, passwords should be used.
- Internet equipment as routers to be protected with password.
- Data that you use to store information which can be financial, or non-financial by encryption.
- Information should be protected in all types of its representation in transmission by encrypting it.

Benefits of Computer Security Awareness

In all this digital world, what is the biggest hole or the weakest point of the security?

Answer. It is us, humans.

Most of the security breaches come from uninformed and untrained persons which give information to a third party or publish data in Internet without knowing the consequences.

So the benefits of computer security awareness are obvious as it directly minimizes the potential of you being hacked off your identity, your computer, your organization.

Potential Losses due to Security Attacks

The potential loses in this cyberspace are many even if you are using a single computer in your room. Here are some examples that have a direct impact on you and on others −

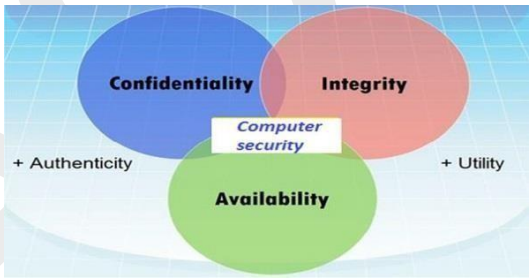
- **Losing you data** − If your computer has been hacked or infected, there is a big chance that all your stored data might be taken by the attacker.

- **Bad usage of your computer resources** − This means that your network or computer can go in overload so you cannot access your genuine services or in a worst case scenario, it can be used by the hacker to attack another machine or network.
- **Reputation loss** − Just think if your Facebook account or business email has been owned by a social engineering attack and it sends fake information to your friends, business partners. You will need time to gain back your reputation.
- **Identity theft** − This is a case where your identity is stolen (photo, name surname, address, and credit card) and can be used for a crime like making false identity documents.

The general state in Computer Security has the ability to detect and prevent attacks and to be able to recover. If these attacks are successful as such then it has to contain the disruption of information and services and check if they are kept low or tolerable.

Different Elements in Computer Security

In order to fulfil these requirements, we come to the three main elements which are **confidentiality, integrity**, and **availability** and the recently added **authenticity and utility**.



Confidentiality

Confidentiality is the concealment of information or resources. Also, there is a need to keep information secret from other third parties that want to have access to it, so just the right people can access it.

Example in real life − Let's say there are two people communicating via an encrypted email they know the decryption keys of each other and they read the email by entering these keys into the email program. If someone else can read these decryption keys when they are entered into the program, then the confidentiality of that email is compromised.

Integrity

Integrity is the trustworthiness of data in the systems or resources by the point of view of preventing unauthorized and improper changes. Generally, Integrity is composed of two sub-elements – data-integrity, which it has to do with the content of the data and authentication which has to do with the origin of the data as such information has values only if it is correct.

Example in real life − Let's say you are doing an online payment of 5 USD, but your information is tampered without your knowledge in a way by sending to the seller 500 USD, this would cost you too much.

In this case cryptography plays a very major role in ensuring data integrity. Commonly used methods to protect data integrity includes hashing the data you receive and comparing it with the hash of the original message. However, this means that the hash of the original data must be provided in a secure way.

Availability

Availability refers to the ability to access data of a resource when it is needed, as such the information has value only if the authorized people can access at right time. Denying access to data nowadays has become a common attack. Imagine a downtime of a live server how costly it can be.

Example in real life − Let's say a hacker has compromised a webserver of a bank and put it down. You as an authenticated user want to do an e-banking transfer but it is impossible to access it, the undone transfer is a money lost for the bank.

The different terminology used in Computer Security.

- **Unauthorized access** − An unauthorized access is when someone gains access to a server, website, or other sensitive data using someone else's account details.
- **Hacker** − Is a Person who tries and exploits a computer system for a reason which can be money, a social cause, fun etc.
- **Threat** − Is an action or event that might compromise the security.
- **Vulnerability** − It is a weakness, a design problem or implementation error in a system that can lead to an unexpected and undesirable event regarding security system.
- **Attack** − Is an assault on the system security that is delivered by a person or a machine to a system. It violates security.

- **Antivirus or Antimalware** − Is a software that operates on different OS which is used to prevent from malicious software.
- **Social Engineering** − Is a technique that a hacker uses to stole data by a person for different for purposes by psychological manipulation combined with social scenes.
- **Virus** − It is a malicious software that installs on your computer without your consent for a bad purpose.
- **Firewall** − It is a software or hardware which is used to filter network traffic based on rules.
- **Risk -** Risk refers to the potential for loss or damage when a threat exploits a vulnerability. Examples of risk include financial losses as a result of business disruption, loss of privacy, reputational damage, legal implications, and can even include loss of life.

  Risk can also be defined as follows:

  Risk = Threat X Vulnerability

QoS (quality of service)

- Quality of service (QoS) refers to any technology that manages data traffic to reduce packet loss, latency and jitter on the network.
- QoS controls and manages network resources by setting priorities for specific types of data on the network.
- Organizations use QoS to meet the traffic requirements of sensitive applications, such as real-time voice and video, and to prevent the degradation of quality caused by packet loss, delay and jitter.
- Organizations can achieve QoS by using certain tools and techniques, such as jitter buffer and traffic shaping.
- For many organizations, QoS is included in the service-level agreement (SLA) with their network service provider to guarantee a certain level of performance.

QoS parameters

Organizations can measure QoS quantitatively by using several parameters, including the following:

- **Packet loss** happens when network links become congested and routers and switches start dropping packets. When packets are dropped during real-time communication, such as a voice or video calls, these sessions can experience jitter and gaps in speech.
- **Jitter** is the result of network congestion, timing drift and route changes. Too much jitter can degrade the quality of voice and video communication.
- **Latency** is the time it takes a packet to travel from its source to its destination. Latency should be as close to zero as possible. If a voice over IP call has a high amount of latency, it can experience echo and overlapping audio.

- **Bandwidth** is the capacity of a network communications link to transmit the maximum amount of data from one point to another in a given amount of time. QoS optimizes the network by managing bandwidth and setting priorities for applications that require more resources than others.
- **Mean opinion score** (MOS) is a metric to rate voice quality that uses a five-point scale, with a five indicating the highest quality.

Implementing QoS

Three models exist to implement QoS: Best Effort, Integrated Services and Differentiated Services.

Best Effort is a QoS model where all the packets receive the same priority and there is no guaranteed delivery of packets. Best Effort is applied when networks have not configured QoS policies or when the infrastructure does not support QoS.

Integrated Services (IntServ) is a QoS model that reserves bandwidth along a specific path on the network. Applications ask the network for resource reservation, and network devices monitor the flow of packets to make sure network resources can accept the packets.

Differentiated Services (DiffServ) is a QoS model where network elements, such as routers and switches, are configured to service multiple classes of traffic with different priorities. Network traffic must be divided into classes based on a company's requirements.

Firewall

A firewall is a network security device that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules.

Firewalls have been a first line of defense in network security for over 25 years. They establish a barrier between secured and controlled internal networks that can be trusted and untrusted outside networks, such as the Internet.

A firewall can be hardware, software, or both.

Types of firewalls

Firewall is categorized into three basic types −

- Packet filter (Stateless & Stateful)
- Application-level gateway
- Circuit-level gateway

These three categories, however, are not mutually exclusive. Modern firewalls have a mix of abilities that may place them in more than one of the three categories.

Stateless& Stateful Packet Filtering Firewall

In this type of firewall deployment, the internal network is connected to the external network/Internet via a router firewall. The firewall inspects and filters data packet-by-packet.

Packet-filtering firewalls allow or block the packets mostly based on criteria such as source and/or destination IP addresses, protocol, source and/or destination port numbers, and various other parameters within the IP header.

The decision can be based on factors other than IP header fields such as ICMP message type, TCP SYN and ACK bits, etc.

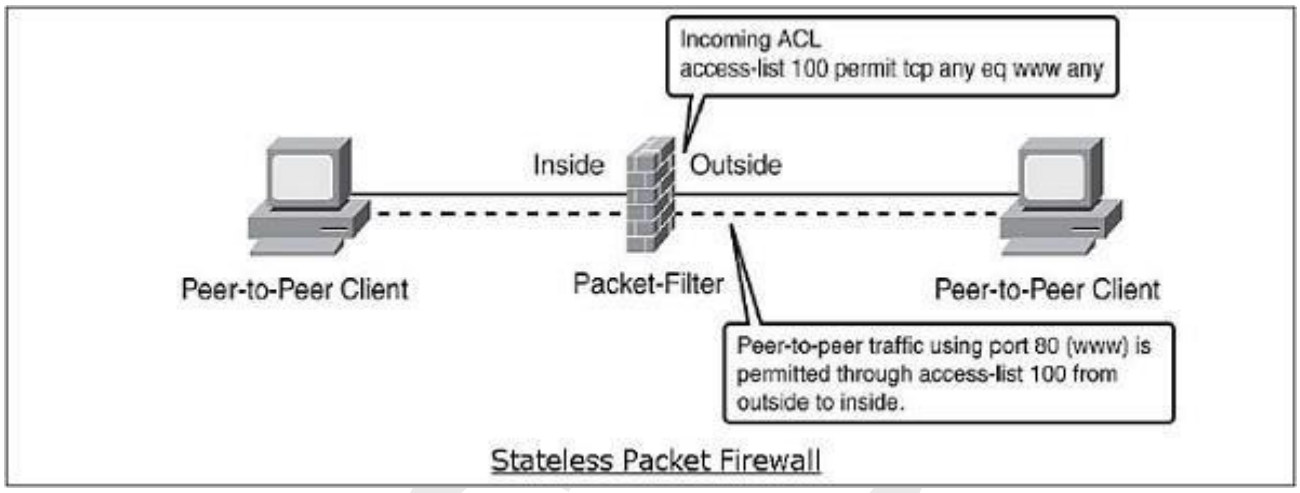
Packet filter rule has two parts −

- **Selection criteria** − It is a used as a condition and pattern matching for decision making.

- **Action field** − This part specifies action to be taken if an IP packet meets the selection criteria. The action could be either block (deny) or permit (allow) the packet across the firewall.

Packet filtering is generally accomplished by configuring Access Control Lists (ACL) on routers or switches. ACL is a table of packet filter rules.

As traffic enters or exits an interface, firewall applies ACLs from top to bottom to each incoming packet, finds matching criteria and either permits or denies the individual packets.



Stateless firewall

- Is a kind of a rigid tool.
- It looks at packet and allows it if its meets the criteria even if it is not part of any established ongoing communication.

Stateful firewalls

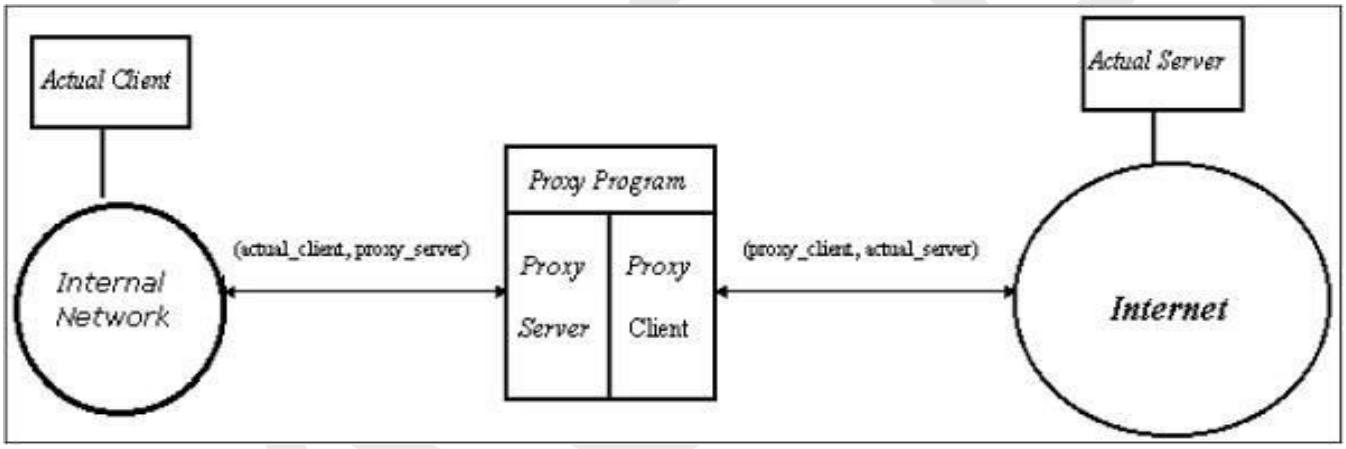
- This type of firewalls offer a more in-depth inspection method over the only ACL based packet inspection methods of stateless firewalls.
- Stateful firewall monitors the connection setup and teardown process to keep a check on connections at the TCP/IP level.
- This allows them to keep track of connections state and determine which hosts have open, authorized connections at any given point in time.
- They reference the rule base only when a new connection is requested. Packets belonging to existing connections are compared to the firewall's state table of open connections, and decision to allow or block is taken.
- This process saves time and provides added security as well. No packet is allowed to trespass the firewall unless it belongs to already established connection.

- It can timeout inactive connections at firewall after which it no longer admit packets for that connection.

Application Gateways

- An application-level gateway acts as a relay node for the application-level traffic.
- They intercept incoming and outgoing packets, run proxies that copy and forward information across the gateway, and function as a ***proxy server***, preventing any direct connection between a trusted server or client and an untrusted host.
- The proxies are application specific.
- They can filter packets at the application layer of the OSI model.

Application-specific Proxies



- An application-specific proxy accepts packets generated by only specified application for which they are designed to copy, forward, and filter. For example, only a Telnet proxy can copy, forward, and filter Telnet traffic.
- If a network relies only on an application-level gateway, incoming and outgoing packets cannot access services that have no proxies configured. For example, if a gateway runs FTP and Telnet proxies, only packets generated by these services can pass through the firewall. All other services are blocked.

Application-level Filtering

- An application-level proxy gateway, examines and filters individual packets, rather than simply copying them and blindly forwarding them across the gateway.
- Application-specific proxies check each packet that passes through the gateway, verifying the contents of the packet up through the application layer. These proxies can filter particular kinds of commands or information in the application protocols.

- Application gateways can restrict specific actions from being performed. For example, the gateway could be configured to prevent users from performing the 'FTP put' command. This can prevent modification of the information stored on the server by an attacker.

Transparent

- Although application-level gateways can be transparent, many implementations require user authentication before users can access an untrusted network, a process that reduces true transparency.
- Authentication may be different if the user is from the internal network or from the Internet.
- For an internal network, a simple list of IP addresses can be allowed to connect to external applications. But from the Internet side a strong authentication should be implemented.
- An application gateway actually relays TCP segments between the two TCP connections in the two directions (Client ↔ Proxy ↔ Server).
- For outbound packets, the gateway may replace the source IP address by its own IP address. The process is referred to as Network Address Translation (NAT). It ensures that internal IP addresses are not exposed to the Internet.

Circuit-Level Gateway

- The circuit-level gateway is an intermediate solution between the packet filter and the application gateway.
- It runs at the transport layer and hence can act as proxy for any application.
- Similar to an application gateway, the circuit-level gateway also does not permit an end-to-end TCP connection across the gateway.
- It sets up two TCP connections and relays the TCP segments from one network to the other. But, it does not examine the application data like application gateway. Hence, sometime it is called as 'Pipe Proxy'.

context, which refers to using information from previous connections and packets belonging to the same connection.

Unified threat management (UTM) firewall

A UTM device typically combines, in a loosely coupled way, the functions of a stateful inspection firewall with intrusion prevention and antivirus. It may also include additional services and often cloud management. UTMs focus on simplicity and ease of use.

Next-generation firewall (NGFW)

Firewalls have evolved beyond simple packet filtering and stateful inspection. Most companies are deploying next-generation firewalls to block modern threats such as advanced malware and application-layer attacks.

A next-generation firewall must include:

- Standard firewall capabilities like stateful inspection
- Integrated intrusion prevention
- Application awareness and control to see and block risky apps
- Upgrade paths to include future information feeds
- Techniques to address evolving security threats

While these capabilities are increasingly becoming the standard for most companies, NGFWs can do more.

Threat-focused NGFW

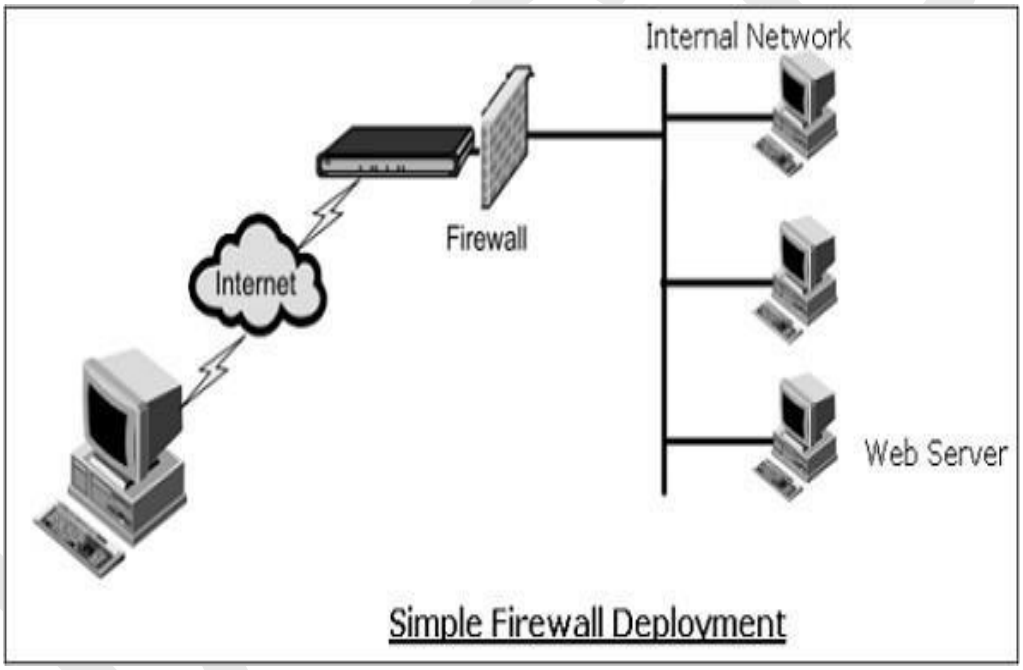
These firewalls include all the capabilities of a traditional NGFW and also provide advanced threat detection and remediation. With a threat-focused NGFW you can:

- Know which assets are most at risk with complete context awareness
- Quickly react to attacks with intelligent security automation that sets policies and hardens your defenses dynamically
- Better detect evasive or suspicious activity with network and endpoint event correlation
- Greatly decrease the time from detection to cleanup with retrospective security that continuously monitors for suspicious activity and behavior even after initial inspection
- Ease administration and reduce complexity with unified policies that protect across the entire attack continuum

Firewall Deployment with DMZ

A firewall is a mechanism used to control network traffic 'into' and 'out' of an organizational internal network. In most cases these systems have two network interfaces, one for the external network such as the Internet and the other for the internal side.

The firewall process can tightly control what is allowed to traverse from one side to the other. An organization that wishes to provide external access to its web server can restrict all traffic arriving at firewall expect for port 80 (the standard http port). All other traffic such as mail traffic, FTP, SNMP, etc., is not allowed across the firewall into the internal network. An example of a simple firewall is shown in the following diagram.

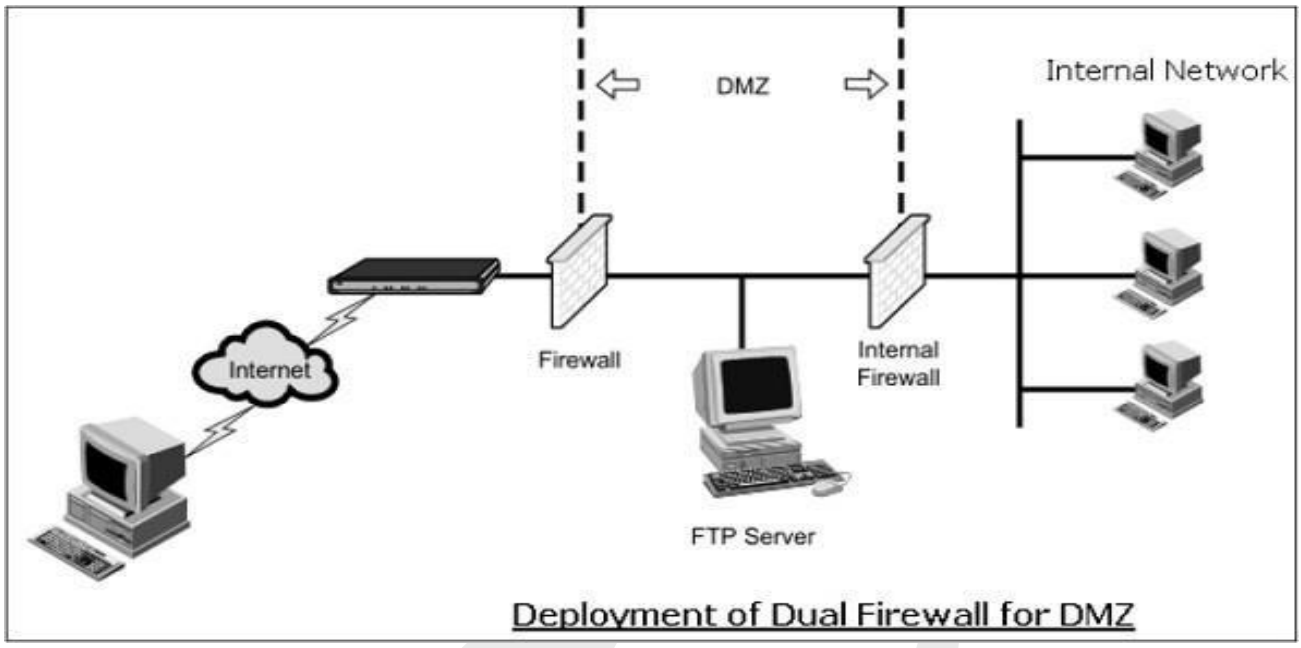


In the above simple deployment, though all other accesses from outside are blocked, it is possible for an attacker to contact not only a web server but any other host on internal network that has left port 80 open by accident or otherwise.

Hence, the problem most organizations face is how to enable legitimate access to public services such as web, FTP, and e-mail while maintaining tight security of the internal network. The typical approach is deploying firewalls to provide a Demilitarized Zone (DMZ) in the network.

In this setup (illustrated in following diagram), two firewalls are deployed; one between the external network and the DMZ, and another between the DMZ and the internal network. All public servers are placed in the DMZ.

With this setup, it is possible to have firewall rules which allow public access to the public servers but the interior firewall can restrict all incoming connections. By having the DMZ, the public servers are provided with adequate protection instead of placing them directly on external network.



Deployment of Dual Firewall for DMZ

IPTABLES

- The Linux kernel comes with a packet filtering framework named netfilter.
- It allows you to allow, drop and modify traffic leaving in and out of a system.
- A tool, iptables builds upon this functionality to provide a powerful firewall, which you can configure by adding rules.
- In addition, other programs such as fail2ban also use iptables to block attackers.

How does iptables work?

- iptables is just a command-line interface to the packet filtering functionality in netfilter.
- The packet filtering mechanism provided by iptables is organized into three different kinds of structures: **tables**, **chains** and **targets**.
- A table is something that allows you to process packets in specific ways. The default table is the *filter* table, although there are other tables too.
- These tables have chains attached to them. These chains allow you to inspect traffic at various points, such as when they just arrive on the network interface or just before they're handed over to a process.
- You can add rules to them match specific packets — such as TCP packets going to port 80 — and associate it with a target.
- A target decides the fate of a packet, such as allowing or rejecting it.
- When a packet arrives (or leaves, depending on the chain), iptables matches it against rules in these chains one-by-one.
- When it finds a match, it jumps onto the target and performs the action associated with it. If it doesn't find a match with any of the rules, it simply does what the **default policy** of the chain tells it to. The default policy is also a target. By default, all chains have a default policy of allowing packets.

Tables

Tables allow you to do very specific things with packets. There are four tables:

- **The filter table:** This is the default and perhaps the most widely used table. It is used to make decisions about whether a packet should be allowed to reach its destination.

- **The mangle table:** This table allows you to alter packet headers in various ways, such as changing TTL values.

- **The *nat* table:** This table allows you to route packets to different hosts on NAT (Network Address Translation) networks by changing the source and destination addresses of packets. It is often used to allow access to services that can't be accessed directly, because they're on a NAT network.

- **The *raw* table:** Iptables is a stateful firewall, which means that packets are inspected with respect to their "state". (For example, a packet could be part of a new connection, or it could be part of an existing connection.) The *raw* table allows you to work with packets before the kernel starts tracking its state. In addition, you can also exempt certain packets from the state-tracking machinery.

In addition, some kernels also have a *security* table. It is used by SELinux to implement policies based on SELinux security contexts.

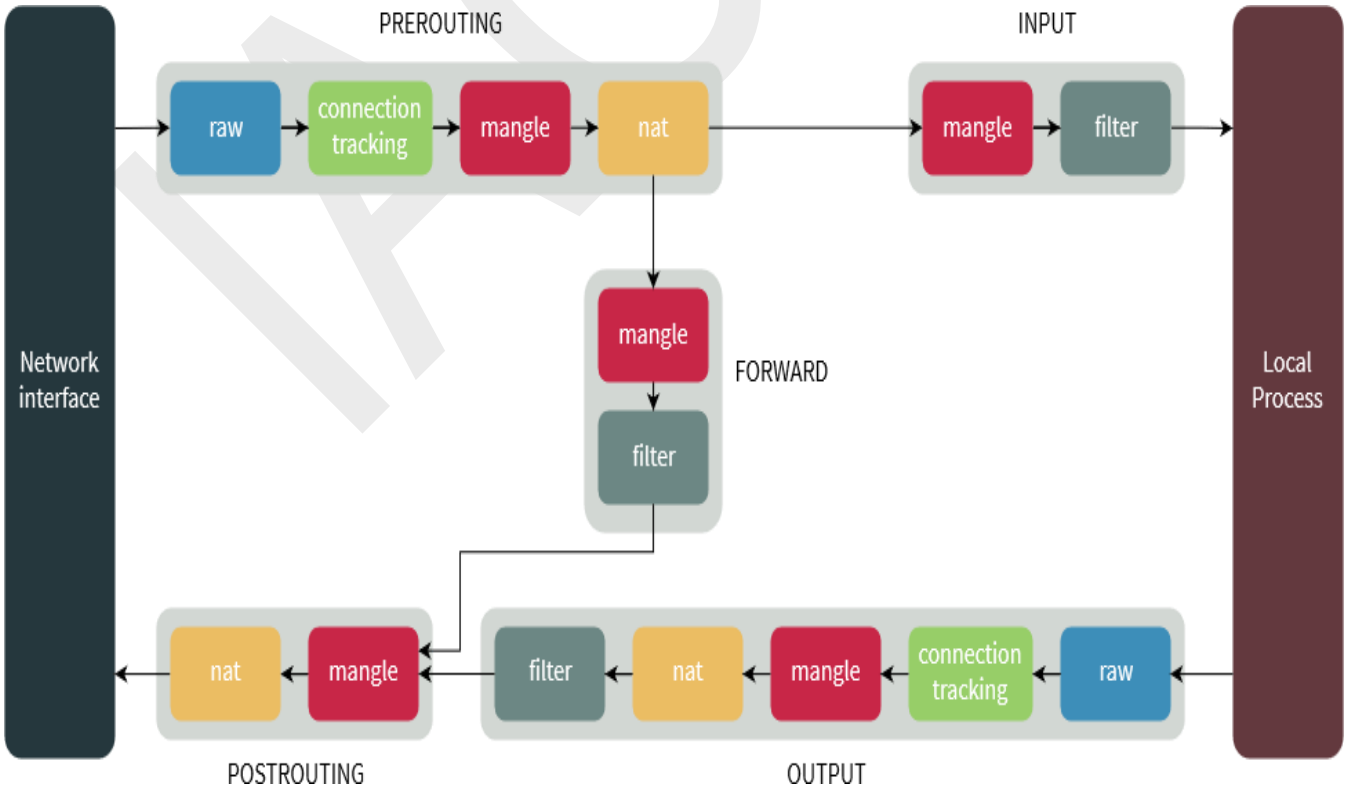
TABLES	USE-CASE
FILTER	Used for Normal Filtering of traffic based on rules defined by the user, like accept, reject etc. This is the highly used table in the iptables firewall. And is very helpful in carrying out normal day to day blocking and filtering.
NAT	Iptable firewall being a matured firewall has capabilities other than normal filtering. Iptable can be used for Network Address Translation Purposes. This table contains rules related to NAT
MANGLE	Rules in this table can be used to modify the packets based on the user given criteria. User can modify the TTL, MSS value, Terms of Service (Like which traffic should be given more priority etc)>
RAW	Primarily used to add No connection tracking Rules.
SECURITY	Used for Mandatory Access Control networking rules

Chains

Each of these tables are composed of a few default chains. These chains allow you to filter packets at various points.

- **The PREROUTING chain:** Rules in this chain apply to packets as they just arrive on the network interface. This chain is present in the *nat*, *mangle* and *raw* tables.
- **The INPUT chain:** Rules in this chain apply to packets just before they're given to a local process. This chain is present in the *mangle* and *filter* tables.
- **The OUTPUT chain:** The rules here apply to packets just after they've been produced by a process. This chain is present in the *raw*, *mangle*, nat and *filter* tables.
- **The FORWARD chain:** The rules here apply to any packets that are routed through the current host. This chain is only present in the *mangle* and *filter* tables.
- **The POSTROUTING chain:** The rules in this chain apply to packets as they just leave the network interface. This chain is present in the *nat* and *mangle* tables.

The diagram below shows the flow of packets through the chains in various tables:



Targets

As we've mentioned before, chains allow you to filter traffic by adding rules to them. So for example, you could add a rule on the filter table's INPUT chain to match traffic on port 22. But what would you do after matching them? That's what targets are for — they decide the fate of a packet.

Some targets are **terminating**, which means that they decide the matched packet's fate immediately. The packet won't be matched against any other rules. The most commonly used terminating targets are:

- **ACCEPT:** This causes iptables to accept the packet.
- **DROP:** iptables drops the packet. To anyone trying to connect to your system, it would appear like the system didn't even exist.
- **REJECT:** iptables "rejects" the packet. It sends a "connection reset" packet in case of TCP, or a "destination host unreachable" packet in case of UDP or ICMP.

On the other hand, there are **non-terminating** targets, which keep matching other rules even if a match was found.

An example of this is the built-in LOG target.

When a matching packet is received, it logs about it in the kernel logs.

However, iptables keeps matching it with rest of the rules too.

Sometimes, you may have a complex set of rules to execute once you've matched a packet. To simplify things, you can create a custom chain.

Then, you can jump to this chain from one of the custom chains.

1. Displaying the Status of Your Firewall

Type the following command as root:

 # iptables -L -n -v

Sample outputs:

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination

Above output indicates that the firewall is not active. The following sample shows an active firewall:
Where,

- **-L** : List rules.
- **-v** : Display detailed information. This option makes the list command show the interface name, the rule options, and the TOS masks. The packet and byte counters are also listed, with the suffix 'K', 'M' or 'G' for 1000, 1,000,000 and 1,000,000,000 multipliers respectively.
- **-n** : Display IP address and port in numeric format. Do not use DNS to resolve names. This will speed up listing.

1.1. To inspect firewall with line numbers, enter:

- **# iptables -n -L -v --line-numbers**

You can use line numbers to delete or insert new rules into the firewall.

1.2. To display INPUT or OUTPUT chain rules, enter:

- **# iptables -L INPUT -n -v**
  # iptables -L OUTPUT -n -v --line-numbers

2. Stop / Start / Restart the Firewall

If you are using CentOS / RHEL / Fedora Linux, enter:

- **# service iptables stop**
- **# service iptables start**

- **# service iptables restart**

You can use the iptables command itself to stop the firewall and delete all rules:

- **# iptables –F**
- **# iptables –X**
- **# iptables -t nat –F**
- **# iptables -t nat –X**
- **# iptables -t mangle –F**
- **# iptables -t mangle –X.**
- **# iptables -P INPUT ACCEPT**
- **# iptables -P OUTPUT ACCEPT**
- **# iptables -P FORWARD ACCEPT**

Where,

- **-F** : Deleting (flushing) all the rules.
- **-X** : Delete chain.
- **-t table_name** : Select table (called nat or mangle) and delete/flush rules.
- **-P** : Set the default policy (such as DROP, REJECT, or ACCEPT).

3. Delete Firewall Rules

To display line number along with other information for existing rules, enter:

- **# iptables -L INPUT -n --line-numbers**
- **# iptables -L OUTPUT -n --line-numbers**
- **# iptables -L OUTPUT -n --line-numbers | less**
- **# iptables -L OUTPUT -n --line-numbers | grep 202.54.1.1**

You will get the list of IP. Look at the number on the left, then use number to delete it. For example delete line number 4, enter:

- **# iptables -D INPUT 4**

OR find source IP 202.54.1.1 and delete from rule:

- **# iptables -D INPUT -s 202.54.1.1 -j DROP**

Where,

- **-D** : Delete one or more rules from the selected chain

4. Insert Firewall Rules

To insert one or more rules in the selected chain as the given rule number use the following syntax. First find out line numbers, enter:

 # iptables -L INPUT -n –line-numbers

Sample outputs:

```
Chain INPUT (policy DROP)
num  target     prot opt source               destination
1    DROP       all --  202.54.1.1          0.0.0.0/0
2    ACCEPT     all --  0.0.0.0/0            0.0.0.0/0           state NEW,ESTABLISHED
```

To insert rule between 1 and 2, enter:

 # iptables -I INPUT 2 -s 202.54.1.2 -j DROP

To view updated rules, enter:

 # iptables -L INPUT -n --line-numbers

Sample outputs:

```
Chain INPUT (policy DROP)
num  target     prot opt source               destination
1    DROP       all --  202.54.1.1          0.0.0.0/0
2    DROP       all --  202.54.1.2          0.0.0.0/0
3    ACCEPT     all --  0.0.0.0/0            0.0.0.0/0           state NEW,ESTABLISHED
```

5. Save Firewall Rules

To save firewall rules under CentOS / RHEL / Fedora Linux, enter:

 # service iptables save

In this example, drop an IP and save firewall rules:

 # iptables -A INPUT -s 202.5.4.1 -j DROP

 # service iptables save

For all other distros use the iptables-save command:

 # iptables-save > /root/my.active.firewall.rules

 # cat /root/my.active.firewall.rules

6. Restore Firewall Rules

To restore firewall rules form a file called /root/my.active.firewall.rules, enter:

 # iptables-restore < /root/my.active.firewall.rules

To restore firewall rules under CentOS / RHEL / Fedora Linux, enter:

 # service iptables restart

7. Set the Default Firewall Policies

To drop all traffic:

 # iptables -P INPUT DROP
 # iptables -P OUTPUT DROP
 # iptables -P FORWARD DROP
 # iptables -L -v -n

you will not able to connect anywhere as all traffic is dropped

 # ping test.com
 # wget http://www.kernel.org/pub/linux/kernel/v3.0/testing/linux-3.2-rc5.tar.bz2

7.1. Only Block Incoming Traffic

To drop all incoming / forwarded packets, but allow outgoing traffic, enter:

 # iptables -P INPUT DROP
 # iptables -P FORWARD DROP
 # iptables -P OUTPUT ACCEPT
 # iptables -A INPUT -m state --state NEW,ESTABLISHED -j ACCEPT
 # iptables -L -v -n

8. Drop Private Network Address On Public Interface

IP spoofing is nothing but to stop the following IPv4 address ranges for private networks on your public interfaces. Packets with non-routable source addresses should be rejected using the following syntax:

 # iptables -A INPUT -i eth1 -s 192.168.0.0/24 -j DROP
 # iptables -A INPUT -i eth1 -s 10.0.0.0/8 -j DROP

9. Blocking an IP Address (BLOCK IP)

To block an attackers ip address called 1.2.3.4, enter:
 # iptables -A INPUT -s 1.2.3.4 -j DROP
 # iptables -A INPUT -s 192.168.0.0/24 -j DROP

10. Block Incoming Port Requests (BLOCK PORT)

To block all service requests on port 80, enter:
 # iptables -A INPUT -p tcp --dport 80 -j DROP
 # iptables -A INPUT -i eth1 -p tcp --dport 80 -j DROP

To block port 80 only for an ip address 1.2.3.4, enter:
 # iptables -A INPUT -p tcp -s 1.2.3.4 --dport 80 -j DROP
 # iptables -A INPUT -i eth1 -p tcp -s 192.168.1.0/24 --dport 80 -j DROP

11. Block Outgoing IP Address

To block outgoing traffic to a particular host or domain such as test.com,

Sample outputs:
test.com has address 75.126.153.206
Note down its ip address and type the following to block all outgoing traffic to 75.126.153.206:
 # iptables -A OUTPUT -d 75.126.153.206 -j DROP

You can use a subnet as follows:

 # iptables -A OUTPUT -d 192.168.1.0/24 -j DROP

 # iptables -A OUTPUT -o eth1 -d 192.168.1.0/24 -j DROP

11.1. Example - Block Facebook.com Domain

First, find out all ip address of facebook.com, enter:

 # host -t a www.facebook.com

Sample outputs:

www.facebook.com has address 69.171.228.40

Find CIDR for 69.171.228.40, enter:

 # whois 69.171.228.40 | grep CIDR

Sample outputs:

CIDR: 69.171.224.0/19

To prevent outgoing access to www.facebook.com, enter:

 # iptables -A OUTPUT -p tcp -d 69.171.224.0/19 -j DROP

You can also use domain name, enter:

 # iptables -A OUTPUT -p tcp -d www.facebook.com -j DROP

 # iptables -A OUTPUT -p tcp -d facebook.com -j DROP

From the iptables man page:

... specifying any name to be resolved with a remote query such as DNS (e.g., facebook.com is a really bad idea), a network IP address (with /mask), or a plain IP address ...

12. Log and Drop Packets

Type the following to log and block IP spoofing on public interface called eth1

 # iptables -A INPUT -i eth1 -s 10.0.0.0/8 -j LOG --log-prefix "IP_SPOOF A: "

 # iptables -A INPUT -i eth1 -s 10.0.0.0/8 -j DROP

By default everything is logged to /var/log/messages file.

 # tail -f /var/log/messages

 # grep --color 'IP SPOOF' /var/log/messages

13. Log and Drop Packets with Limited Number of Log Entries

The -m limit module can limit the number of log entries created per time. This is used to prevent flooding your log file. To log and drop spoofing per 5 minutes, in bursts of at most 7 entries .

 # iptables -A INPUT -i eth1 -s 10.0.0.0/8 -m limit --limit 5/m --limit-burst 7 -j LOG --log-prefix "IP_SPOOF A: "

 # iptables -A INPUT -i eth1 -s 10.0.0.0/8 -j DROP

14. Drop or Accept Traffic From Mac Address

Use the following syntax:

 # iptables -A INPUT -m mac --mac-source 00:0F:EA:91:04:08 -j DROP

*only accept traffic for TCP port # 8080 from mac 00:0F:EA:91:04:07 *

 # iptables -A INPUT -p tcp --destination-port 22 -m mac --mac-source 00:0F:EA:91:04:07 -j ACCEPT

15. Block or Allow ICMP Ping Request

Type the following command to block ICMP ping requests:

 # iptables -A INPUT -p icmp --icmp-type echo-request -j DROP

 # iptables -A INPUT -i eth1 -p icmp --icmp-type echo-request -j DROP

Ping responses can also be limited to certain networks or hosts:

 # iptables -A INPUT -s 192.168.1.0/24 -p icmp --icmp-type echo-request -j ACCEPT

The following only accepts limited type of ICMP requests:

** assumed that default INPUT policy set to DROP **

iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT

iptables -A INPUT -p icmp --icmp-type destination-unreachable -j ACCEPT

iptables -A INPUT -p icmp --icmp-type time-exceeded -j ACCEPT

** all our server to respond to pings **

iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT

16. Open Range of Ports

Use the following syntax to open a range of ports:

iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 7000:7010 -j ACCEPT

17. Open Range of IP Addresses

Use the following syntax to open a range of IP address:

only accept connection to tcp port 80 (Apache) if ip is between 192.168.1.100 and 192.168.1.200

iptables -A INPUT -p tcp --destination-port 80 -m iprange --src-range 192.168.1.100-192.168.1.200 -j ACCEPT

nat example

iptables -t nat -A POSTROUTING -j SNAT --to-source 192.168.1.20-192.168.1.25

18. Established Connections and Restarting The Firewall

When you restart the iptables service it will drop established connections as it unload modules from the system under RHEL / Fedora / CentOS Linux.

Edit, /etc/sysconfig/iptables-config and set IPTABLES_MODULES_UNLOAD as follows:

IPTABLES_MODULES_UNLOAD = no

19. Help Iptables Flooding My Server Screen

Use the crit log level to send messages to a log file instead of console:

iptables -A INPUT -s 1.2.3.4 -p tcp --destination-port 80 -j LOG --log-level crit

20. Block or Open Common Ports

The following shows syntax for opening and closing common TCP and UDP ports:

Replace ACCEPT with DROP to block port:

open port ssh tcp port 22

iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT

iptables -A INPUT -s 192.168.1.0***/***24 -m state --state NEW -p tcp --dport 22 -j ACCEPT

open cups (printing service) udp/tcp port 631 for LAN users

iptables -A INPUT -s 192.168.1.0***/***24 -p udp -m udp --dport 631 -j ACCEPT

iptables -A INPUT -s 192.168.1.0***/***24 -p tcp -m tcp --dport 631 -j ACCEPT

allow time sync via NTP for lan users (open udp port 123)

iptables -A INPUT -s 192.168.1.0***/***24 -m state --state NEW -p udp --dport 123 -j ACCEPT

open tcp port 25 (smtp) for all

iptables -A INPUT -m state --state NEW -p tcp --dport 25 -j ACCEPT

 # open dns server ports for all ##

iptables -A INPUT -m state --state NEW -p udp --dport 53 -j ACCEPT

iptables -A INPUT -m state --state NEW -p tcp --dport 53 -j ACCEPT

open http/https (Apache) server port to all

iptables -A INPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT

iptables -A INPUT -m state --state NEW -p tcp --dport 443 -j ACCEPT

open tcp port 110 (pop3) for all

iptables -A INPUT -m state --state NEW -p tcp --dport 110 -j ACCEPT

open tcp port 143 (imap) for all

iptables -A INPUT -m state --state NEW -p tcp --dport 143 -j ACCEPT

open access to Samba file server for lan users only

iptables -A INPUT -s 192.168.1.0***/***24 -m state --state NEW -p tcp --dport 137 -j ACCEPT

iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 138 -j ACCEPT
iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 139 -j ACCEPT
iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 445 -j ACCEPT

open access to proxy server for lan users only
iptables -A INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 3128 -j ACCEPT

open access to mysql server for lan users only
iptables -I INPUT -p tcp --dport 3306 -j ACCEPT

21. Restrict the Number of Parallel Connections To a Server Per Client IP

You can use connlimit module to put such restrictions. To allow 3 ssh connections per client host, enter:

 # iptables -A INPUT -p tcp --syn --dport 22 -m connlimit --connlimit-above 3 -j REJECT

Set HTTP requests to 20:

 # iptables -p tcp --syn --dport 80 -m connlimit --connlimit-above 20 --connlimit-mask 24 -j DROP

Where,

1. **--connlimit-above 3** : Match if the number of existing connections is above 3.
2. **--connlimit-mask 24** : Group hosts using the prefix length. For IPv4, this must be a number between (including) 0 and 32.

22. List NAT rules

The syntax is

iptables -t nat -L -n -v

Another option:

iptables -t nat -v -L -n --line-number

23. Delete NAT rules

The syntax is as follows to list NAT rules on Linux:

iptables -t nat -v -L -n --line-number
iptables -t nat -v -L PREROUTING -n --line-number
iptables -t nat -v -L POSTROUTING -n --line-number

To delete PREROUTING rule, run:
iptables -t nat -D PREROUTING {number-here}
iptables -t nat -D PREROUTING 42

To delete POSTROUTING rule, run:
iptables -t nat -D POSTROUTING {number-here}
iptables -t nat -D POSTROUTING 42

24. How to redirect port AA to BB

The syntax is as follows:
iptables -t nat -A PREROUTING -i $interfaceName -p tcp --dport $srcPortNumber -j REDIRECT --to-port $dstPortNumber

To redirect all incoming traffic on port 80 redirect to port 8080
 # iptables -t nat -I PREROUTING --src 0/0 --dst 192.168.1.5 -p tcp --dport 80 -j REDIRECT --to-ports 8080

25. How to reset packet counters

To see iptables counters run:
iptables -L -n -v

To clear/reset the counters for all rules:
iptables -Z
iptables -L -n -v

To reset the counters for INPUT chain only:

iptables -Z INPUT

To reset the counters for rule # 13 in the INPUT chain only:

iptables -Z INPUT 13

26. HowTO: Use iptables Like a Pro

For more information about iptables, please see the manual page by typing man iptables from the command line:

$ man iptables

You can see the help using the following syntax too:

iptables -h

To see help with specific commands and targets, enter:

iptables -j DROP -h

27. Testing Your Firewall

Find out if ports are open or not, enter:

netstat -tulpn

Find out if tcp port 80 open or not, enter:

netstat -tulpn | grep :80

If port 80 is not open, start the Apache, enter:

service httpd start

Make sure iptables allowing access to the port 80:

iptables -L INPUT -v -n | grep 80

Otherwise open port 80 using the iptables for all users:

iptables -A INPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT

service iptables save

Linux Software Firewall(ClearOS Pfsense)

Pfsense and ClearOS are two of the best and highest ranked firewall systems in the market todayBoth Pfsense and ClearOS were carefully and systematically developed to give the best firewall experience to their users. They have fairly similar features including management and creation of rule based policies for managers and users, depending on the preference of the network administrator, as well as the ability to allocate bandwidth in the most optimized ratio to the users and their applications. Neither of them has a customized signature feature, which would have been a huge plus to ensure maximum protection of the network. It would be safe to say that they are fairly similar, but they also have a few differences, the major one being that Pfsense is free and open source while ClearOS is a paid software.

What are the differences between Pfsense and ClearOS?

Feature	Pfsense	ClearOS (Winner)
Operating system	Free BSD	Linux
Pricing	Free and open source	Has free and paid packages
Feature blocking HTTPS and HTTP traffic with modifiable rules	Yes	No
Management policy based time-quota	Yes	No
A comprehensive IPS and full protection against known malwares	Yes	Yes
Support for site to site VPN client	No	Yes
Rule-based bandwidth allocation	Yes	Yes

Pfsense Vs ClearOS- How do they compare?

Additional Features

This is demonstrated as:

- ***Content filtering***– Both firewalls have rule based filtering for keywords, URLs, ports and browsers, which is essential for the key performance of a firewall. However, Pfsense goes above and beyond your average content filtering features to include a management policy for that filters both users and content. It also has a special feature that blocks both HTTP and HTTPS traffic, based on pre-set modifiable rules.
- ***Security protection and intrusion-***They have all the same features here which include a large gateway antivirus to protect the devices against known malware attacks and the ability to create and manage IP table rules. They are also lacking in the same capacity as neither has a customized signature feature.
- ***Remote Connectivity***– Here, once again they are fairly similar except for the fact that pfsense does not support site to site VPN clients.

Both firewalls have great additional features but those in Pfsense make it more flexible and efficient.

Cost and availability

Pfsense is completely free and open source, which makes it openly available on the internet. ClearOS on the other hand, has three packages, the free community package, the home package and the slightly more expensive business package. The packages are ranked according to the number and nature of firewall features from the most basic to the most advanced.

This makes them equally available for use. The pricing model for ClearOS is preferable because it offers the user the choice between three packages with specialized and advanced features, with customer care support for all, even the free community package

Ease of use and configuration

Both pfsense and ClearOS combine the use of a Command Line Interface with a Graphical User interface. However, for pfsense, this is only the case during the installation process, so we can say that Pfsense has a GUI, while ClearOS combines both. This implies that ClearOS is more effective. In addition to this, ClearOS is outlaid in a way that makes it one of the sleekest and easiest to use firewall software, as well as very visually appealing.

ClearOS is definitely the easier to use and more efficient software of the two.

Pfsense Vs ClearOS- A comparison review

Pfsense

Pfsense is a free BSD based software that is specifically designed to perform firewall tasks as well as routing. It has quite a number of built in features which include but are not limited to Dynamic DNS, multi-WAN support, different authentication methods and a hardware failover feature. One of its rather convenient features is the captive portal one which enables all DNS queries to be resolved to one IP address, which could be a landing page for a hotspot.

It typically runs on quite the range of hardware, but mainly supports devices with x86 architecture. However, their website has an elaborate firmware guide which aids the user in choosing a compatible device. The installation process is command line based, but fairly simple because of the help of the set-up assistant. The set-up assistant also allows you to assign interfaces to your preference during the installation, which is very handy, given that the main board may be a bit difficult to navigate for a new user.

Pros

- Has a large number of additional features
- It is free and open source
- It has an additional use as a router

Cons

- It may be fairly difficult to navigate for a new user

ClearOS

ClearOS is a Linux based firewall software that is available in three different versions; the free community version, the paid home version and the paid business version. The developers pride themselves on the amount of care and time that has been taken to develop the user interface, making it one of the sleekest and easiest to use firewall software available. It combines both GUI and CLI in the user's preference. This implies that it can run in GUI for a new or inexperienced user, and in CLI for a stereotypical geek or someone looking to learn or gain experience with a CLI, either way, it is good for both. It is offered on ClearBOX devices, which ensure high quality hardware.

Pros

- It is extremely user friendly
- The version purchased can be expanded to meet user needs

Cons

- It is primarily a paid product

NGINX

NGINX is open source software for web serving, reverse proxying, caching, load balancing, media streaming, and more. It started out as a web server designed for maximum performance and stability. In addition to its HTTP server capabilities, NGINX can also function as a proxy server for email (IMAP, POP3, and SMTP) and a reverse proxy and load balancer for HTTP, TCP, and UDP servers.

Backstory

Igor Sysoev originally wrote NGINX to solve the C10K problem, a term coined in 1999 to describe the difficulty that existing web servers experienced in handling large numbers (the *10K*) of concurrent connections (the *C*). With its event-driven, asynchronous architecture, NGINX revolutionized how servers operate in high-performance contexts and became the fastest web server available.

After open sourcing the project in 2004 and watching its use grow exponentially, Sysoev co-founded NGINX, Inc. to support continued development of NGINX and to market NGINX Plus as a commercial product with additional features designed for enterprise customers. Today, NGINX and NGINX Plus can handle hundreds of thousands of concurrent connections, and power more than 50% of the busiest sites on the web.

NGINX as a Web Server

The goal behind NGINX was to create the fastest web server around, and maintaining that excellence is still a central goal of the project. NGINX consistently beats Apache and other servers in benchmarks measuring web server performance. Since the original release of NGINX however, websites have expanded from simple HTML pages to dynamic, multifaceted content. NGINX has grown along with it and now supports all the components of the modern Web, including WebSocket, HTTP/2, and streaming of multiple video formats (HDS, HLS, RTMP, and others).

NGINX Beyond Web Serving

Though NGINX became famous as the fastest web server, the scalable underlying architecture has proved ideal for many web tasks beyond serving content. Because it can handle a high

volume of connections, NGINX is commonly used as a reverse proxy and load balancer to manage incoming traffic and distribute it to slower upstream servers – anything from legacy database servers to microservices.

NGINX also is frequently placed between clients and a second web server, to serve as an SSL/TLS terminator or a web accelerator. Acting as an intermediary, NGINX efficiently handles tasks that might slow down your web server, such as negotiating SSL/TLS or compressing and caching content to improve performance. Dynamic sites, built using anything from Node.js to PHP, commonly deploy NGINX as a content cache and reverse proxy to reduce load on application servers and make the most effective use of the underlying hardware.

How Does Nginx Work?

Nginx is built to offer **low memory usage** and high concurrency. Rather than creating new processes for each web request, Nginx uses an asynchronous, event-driven approach where requests are handled in a single thread.

With Nginx, one master process can control multiple worker processes. The master maintains the worker processes, while the workers do the actual processing. Because Nginx is asynchronous, each request can be executed by the worker concurrently without blocking other requests.

Some common features seen in Nginx include:

- Reverse proxy with caching
- IPv6
- Load balancing
- FastCGI support with caching
- WebSockets
- Handling of static files, index files, and auto-indexing
- TLS/SSL with SNI

Nginx vs Apache

Nginx was written with an explicit goal of outperforming the Apache web server.Out of the box, serving static files, Nginx uses much less memory than Apache, and can handle roughly four times as many requests per second. However, this performance boost comes at a cost of

decreased flexibility, such as the ability to override systemwide access settings on a per-file basis (Apache accomplishes this with an .htaccess file, while Nginx has no such feature built in)

Nginx also has a reputation of being harder to install and configure than Apache.Formerly, adding third-party modules to Nginx required recompiling the application from source with the modules statically linked. This was partially overcome in version 1.9.11 in February 2016, with the addition of dynamic module loading. However, the modules still must be compiled at the same time as Nginx, and not all modules are compatible with this system; some require the older static linking process.Nginx is generally considered to be less stable on Windows Server than it is on Linux, while Apache has equal support for both.

The way nginx and its modules work is determined in the configuration file. By default, the configuration file is named nginx.conf and placed in the directory /usr/local/nginx/conf, /etc/nginx, or /usr/local/etc/nginx.

Starting, Stopping, and Reloading Configuration

To start nginx, run the executable file. Once nginx is started, it can be controlled by invoking the executable with the -s parameter. Use the following syntax:

nginx -s *signal*

Where *signal* may be one of the following:

- stop — fast shutdown
- quit — graceful shutdown
- reload — reloading the configuration file
- reopen — reopening the log files

For example, to stop nginx processes with waiting for the worker processes to finish serving current requests, the following command can be executed:

nginx -s quit

This command should be executed under the same user that started nginx.

Changes made in the configuration file will not be applied until the command to reload configuration is sent to nginx or it is restarted.

To reload configuration, execute:

nginx -s reload

Once the master process receives the signal to reload configuration, it checks the syntax validity of the new configuration file and tries to apply the configuration provided in it. If this is a success, the master process starts new worker processes and sends messages to old worker processes, requesting them to shut down. Otherwise, the master process rolls back the changes and continues to work with the old configuration. Old worker processes, receiving a command to shut down, stop accepting new connections and continue to service current requests until all such requests are serviced. After that, the old worker processes exit.

A signal may also be sent to nginx processes with the help of Unix tools such as the kill utility. In this case a signal is sent directly to a process with a given process ID. The process ID of the nginx master process is written, by default, to the nginx.pid in the directory /usr/local/nginx/logs or /var/run. For example, if the master process ID is 1628, to send the QUIT signal resulting in nginx's graceful shutdown, execute:

kill -s QUIT 1628

For getting the list of all running nginx processes, the ps utility may be used, for example, in the following way:

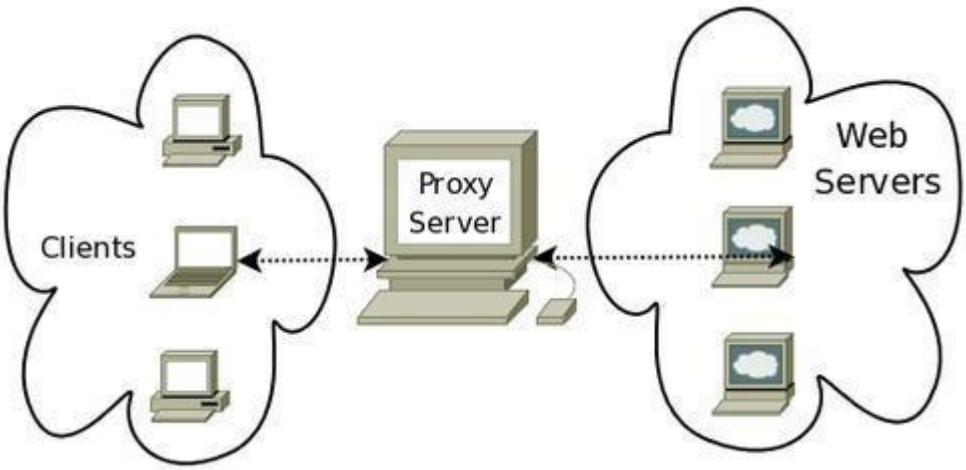
ps -ax | *grep nginx*

Proxy server

A proxy server is a computer system sitting between the client requesting a web document and the target server (another computer system) serving the document. In its simplest form, a proxy server facilitates communication between client and target server without modifying requests or replies. When we initiate a request for a resource from the target server, the proxy server hijacks our connection and represents itself as a client to the target server, requesting the resource on our behalf. If a reply is received, the proxy server returns it to us, giving a feel that we have communicated with the target server.

In advanced forms, a proxy server can filter requests based on various rules and may allow communication only when requests can be validated against the available rules. The rules are

generally based on an IP address of a client or target server, protocol, content type of web documents, web content type, and so on.



As seen in the preceding image, clients can't make direct requests to the web servers. To facilitate communication between clients and web servers, we have connected them using a proxy server which is acting as a medium of communication for clients and web servers.

Sometimes, a proxy server can modify requests or replies, or can even store the replies from the target server locally for fulfilling the same request from the same or other clients at a later stage. Storing the replies locally for use at a later time is known as **caching**. Caching is a popular technique used by proxy servers to save bandwidth, empowering web servers, and improving the end user's browsing experience.

Proxy servers are mostly deployed to perform the following:

- Reduce bandwidth usage
- Enhance the user's browsing experience by reducing page load time which, in turn, is achieved by caching web documents
- Enforce network access policies
- Monitoring user traffic or reporting Internet usage for individual users or groups
- Enhance user privacy by not exposing a user's machine directly to Internet
- Distribute load among different web servers to reduce load on a single server
- Empower a poorly performing web server
- Filter requests or replies using an integrated virus/malware detection system
- Load balance network traffic across multiple Internet connections
- Relay traffic around within a local area network

In simple terms, a proxy server is an agent between a client and target server that has a list of rules against which it validates every request or reply, and then allows or denies access accordingly.

Reverse proxy

Reverse proxying is a technique of storing the replies or resources from a web server locally so that the subsequent requests to the same resource can be satisfied from the local copy on the proxy server, sometimes without even actually contacting the web server. The proxy server or web cache checks if the locally stored copy of the web document is still valid before serving the cached copy.

The life of the locally stored web document is calculated from the additional HTTP headers received from the web server. Using HTTP headers, web servers can control whether a given document/response should be cached by a proxy server or not.

Web caching is mostly used:

- To reduce bandwidth usage. A large number of static web documents like CSS and JavaScript files, images, videos, and so on can be cached as they don't change frequently and constitutes the major part of a response from a web server.
- By ISPs to reduce average page load time to enhance browsing experience for their customers on Dial-Up or broadband.
- To take a load off a very busy web server by serving static pages/documents from a proxy server's cache.

Basic functionality of Squid proxy server

After a Squid proxy server is installed, web browsers can be configured to use it as a proxy HTTP server, allowing Squid to retain copies of the documents returned, which, on repeated requests for the same documents, can reduce access time as well as bandwidth consumption. This is often useful for Internet service providers to increase speed to their customers, and LANs that share an Internet connection. Because the caching servers are controlled by the web service operator, caching proxies do not anonymize the user and should not be confused with anonymizing proxies.

A client program (e.g. browser) either has to specify explicitly the proxy server it wants to use (typical for ISP customers), or it could be using a proxy without any extra configuration: "transparent caching", in which case all outgoing HTTP requests are intercepted by Squid and all responses are cached. The latter is typically a corporate set-up (all clients are on the same LAN) and often introduces the privacy concerns mentioned above.

Squid has some features that can help anonymize connections, such as disabling or changing specific header fields in a client's HTTP requests. Whether these are set, and what they are set to do, is up to the person who controls the computer running Squid. People requesting pages through a network which transparently uses Squid may not know whether this information is being logged.[

Squid - Proxy Server

Squid is a full-featured web proxy cache server application which provides proxy and cache services for Hyper Text Transport Protocol (HTTP), File Transfer Protocol (FTP), and other popular network protocols. Squid can implement caching and proxying of Secure Sockets Layer (SSL) requests and caching of Domain Name Server (DNS) lookups, and perform transparent caching. Squid also supports a wide variety of caching protocols, such as Internet Cache Protocol (ICP), the Hyper Text Caching Protocol (HTCP), the Cache Array Routing Protocol (CARP), and the Web Cache Coordination Protocol (WCCP).

The Squid proxy cache server is an excellent solution to a variety of proxy and caching server needs, and scales from the branch office to enterprise level networks while providing extensive, granular access control mechanisms, and monitoring of critical parameters via the Simple Network Management Protocol (SNMP). When selecting a computer system for use as a dedicated Squid caching proxy server for many users ensure it is configured with a large amount of physical memory as Squid maintains an in-memory cache for increased performance.

UTM

What is unified threat management?

Unified threat management (UTM) provides multiple security features and services in a single device or service on the network, protecting users from security threats in a simplified way. UTM includes functions such as anti-virus, anti-spam, content filtering, and web filtering.

How Does UTM Protect Users and Networks?

IT teams are constantly faced with the challenge of protecting their companies' productivity and digital assets against evolving and sophisticated threats, including spam and phishing attacks, viruses, trojans and spyware infected files, unapproved website access, and unapproved content. They have to address these challenges with limited budgets and resources. Having multiple separate devices, each designed to perform a specialized function such as spam filtering, web content filtering, or antivirus protection does not make this task easier. Rather, it adds to the cost and complexity of managing multiple boxes and multiple operating systems.

UTM is a single system that provides the answer to all of these challenges and more:

- It secures the network from viruses, malware, or malicious attachments by scanning the incoming data using Deep Packet Inspection.
- It prevents attacks before they enter the network by inspecting the packet headers.
- It prevents access to unwanted websites by installing enhanced web filtering.
- It provides ability to update automatically with the latest security updates, anti-virus definitions, and new features so that minimal manual intervention is required beyond initial set-up.
- It enables administrators to manage a wide range of security functions with a single management console.

Figure shows how UTM is deployed in an organization's network.

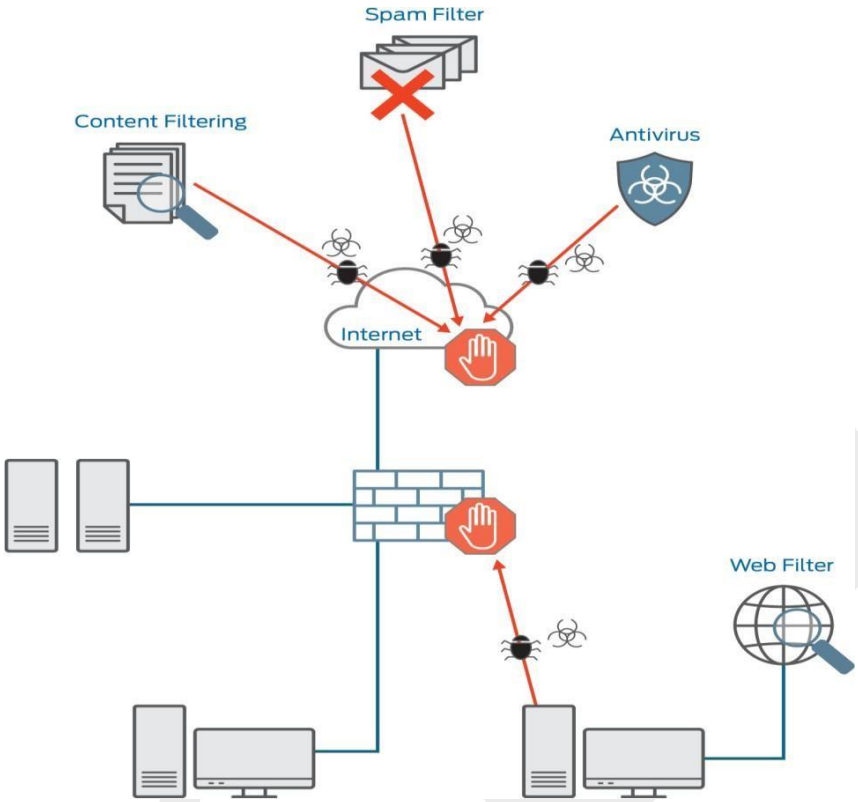


Figure : UTM Deployment in Network

What is a UTM Firewall?

In the early days of network security, a firewall merely filtered traffic based on ports & IP addresses. Over time, firewalls continued to evolve by keeping track of the state of network connections passing through the appliance, which we call "stateful." Unfortunately, cyber threats also evolved & diversified to meet these new challenges, organizations deployed multiple appliances, each with differing roles to defend against different classes of attacks:

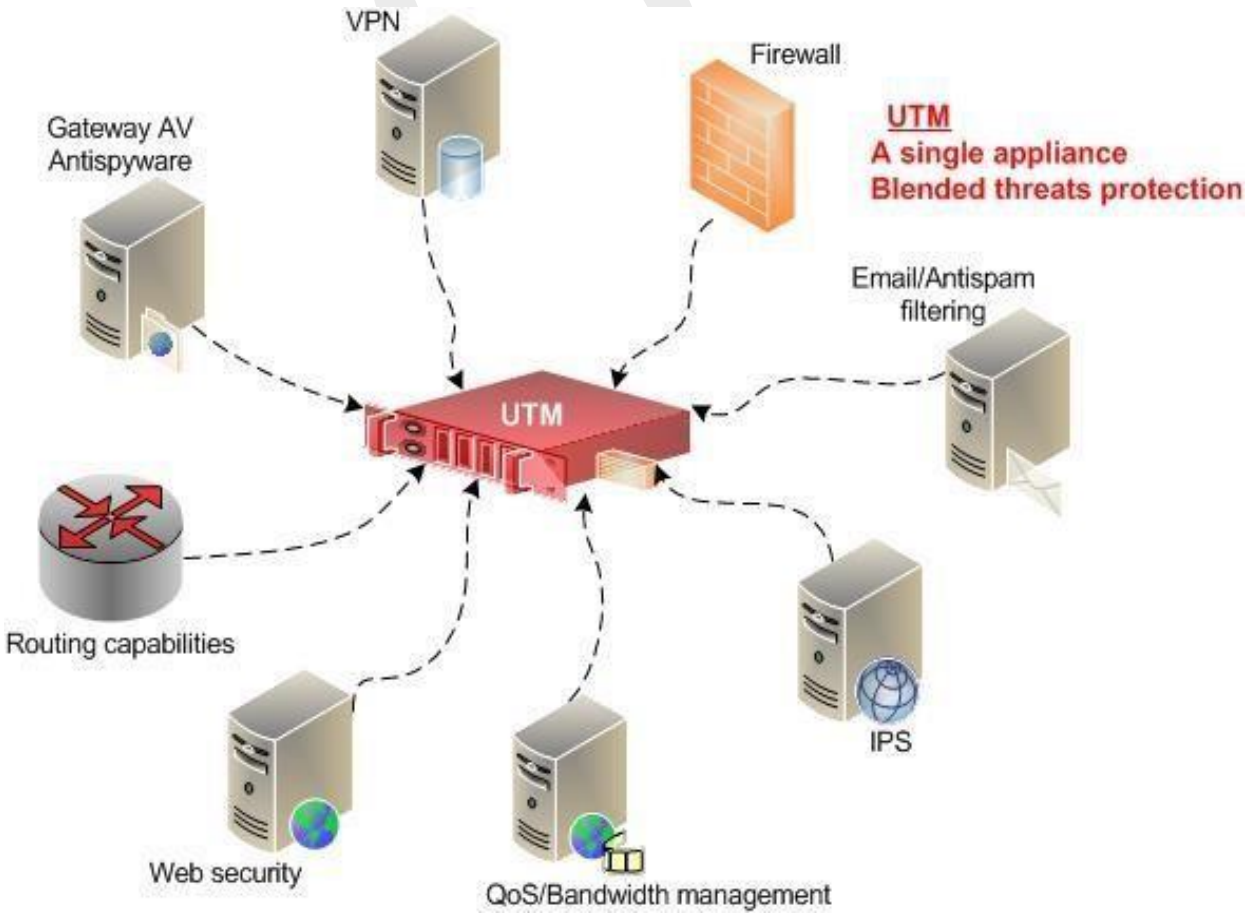
- A stateful packet inspection firewall allowed inbound & outbound traffic on the network
- An additional web proxy filtered content & URLs while scanning with antivirus services
- A separate Intrusion Prevention System (IPS) was often deployed to detect & block malicious traffic
- An appliance was needed for spam filtering to filter junk emails & phishing attempts
- VPN servers connected remote offices or allowed remote users to access company resources

As more & more threats evolved, the industry minted new types of appliances & services to meet the challenge, the traditional stateful appliance approach just could not easily scale to keep up with growing businsses. The complexity became too difficult to manage as network security setups began to resemble Frankenstein's monster with multiple bits & pieces everywhere in an attempt to cover every attack vector. Clearly, a new approach was needed.

The Introduction of UTM

In 2003, several vendors launched "all-in-one" security products to resolve this issue. IDC coined a new term for this new class of firewalls: UTM. By 2004 the name stuck & is still in use to this day.

UTM firewalls could now collocate multiple security services into one appliance, providing robust network protection against a plurality of attack types. To make managing & reporting easier on admins, manufacturers adopted all-in-one management interfaces where multiple services, features, policies, & rules could be centrally managed. Organizations no longer needed to deploy multiple devices--each with its own management dashboard & login credentials--to defend against blended threats.



Benefits that UTM provided

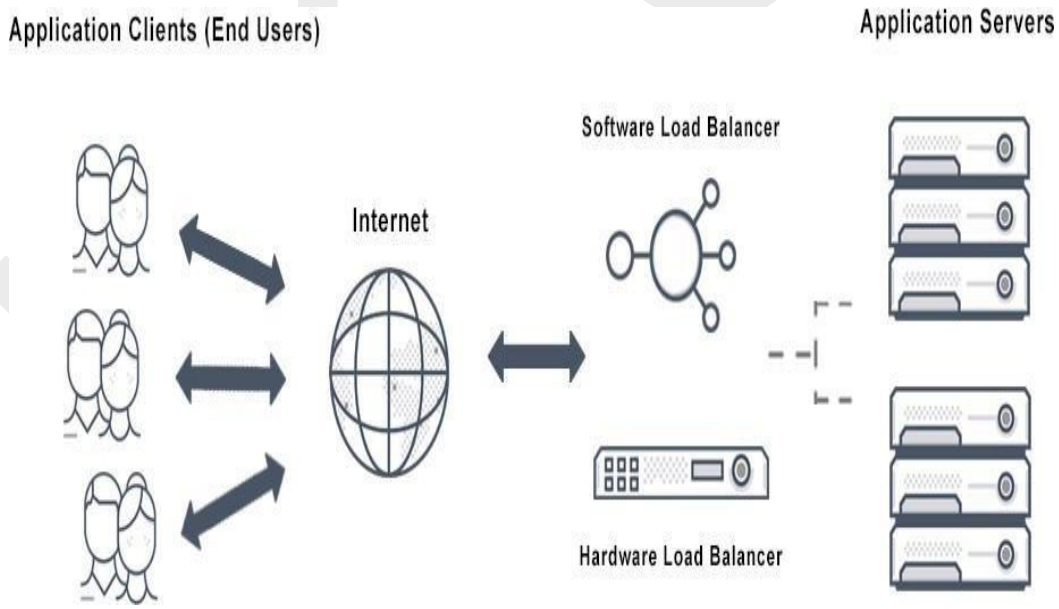
- UTM firewalls protected inbound & outbound traffic from a multitude of threats & attack types
- Antivirus, anti-malware, & anti-spyware services could run concurrently to prevent attacks at the gateway
- Integrated Intrusion Prevention blocked the exploit of vulnerabilities
- Email filtering blocked unwanted emails like spam & email-borne threats
- Web sites & web content could be filtered & monitored from the same central command dashboard
- Control & visibility over traffic flows improved with Quality of Service enhancements & bandwidth management
- Working remotely became more convenient with the ability to connect easily to remote locations with a site-to-site VPN
- Simplification of complex networks allowed for dynamic routing, policy-based routing, & multiple Internet connections on a single secure network

The introduction of UTM firewalls drove an increase in overall network security & reduced complexity & cost, making advanced security options more accessible to small businesses or other SMBs where recruiting expensive IT talent could prove impossible.

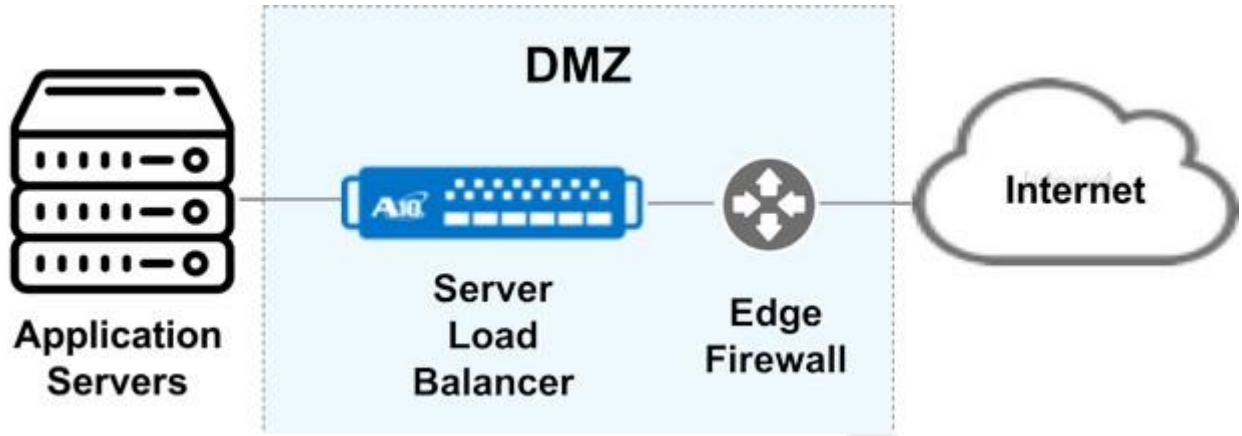
Server Load Balancing

Server Load Balancing (SLB) is a technology that distributes high traffic sites among several servers using a network-based hardware or software-defined appliance. And when load balancing across multiple geo locations, the intelligent distribution of traffic is referred to as global server load balancing (GSLB). The servers can be on premises in a company's own data centers, or hosted in a private cloud or the public cloud. Server load balancers intercepts traffic for a website and reroutes that traffic to servers.

Server load balancing (SLB) is a data center architecture that distributes network traffic evenly across a group of servers. The distributed workloads ensure application availability, scale-out of server resources and health management of server and application systems.



Server Load Balancer systems are often located between the Internet edge routers or firewalls inside the DMZ security zone and the Internet facing application servers.



Server Load Balancer Typical Configuration

In this configuration, the SLB systems act as a reverse proxy, presenting the hosted services to remote network clients. Remote clients over the Internet connect to the SLB system which masquerades as a single application server, then forwards a connection the optimal application server.

What is Server Load Balancing?

Server Load Balancing (SLB) provides network services and content delivery using a series of load balancing algorithms. It prioritizes responses to the specific requests from clients over the network. Server load balancing distributes client traffic to servers to ensure consistent, high-performance application delivery.

Server load balancing ensures application delivery, scalability, reliability and high availability.

How does Server Load Balancing Work?

Server load balancing works within two main types of load balancing:

• Transport-level load balancing is a DNS-based approach which acts independently of the application payload.
• Application-level load balancing uses traffic load to make balancing decisions such as with windows server load balancing.

What are the Advantages of Server Load Balancing?

Distributing incoming network traffic through web server load balancers across multiple servers aims to increase efficiency of application delivery to end users for a reliable application experience. IT teams are increasingly relying on server load balancers to:

• Increase Scalability: load balancers are able to spin up or down server resources based on spikes in traffic to the pool of servers that are best suited to handle these increases in traffic and keep applications performance optimized.

• Redundancy: Using multiple web servers to deliver applications or websites provides a safeguard against the inevitable hardware failure and application downtime. When server load balancers are in place they can automatically transfer traffic to working servers from servers that go down with little to no impact on the end user.

• Maintenance and Performance: Business with web servers distributed across multiple locations and a variety of cloud environments can schedule maintenance at any time to improve performance with minimal impact on application uptime as server load balancers can redirect traffic to resources that are not undergoing maintenance.

What is the Difference Between HTTP Server Load Balancing and TCP Load Balancing?

HTTP server load balancing is a simple HTTP request/response architecture for HTTP traffic. But a TCP load balancer is for applications that do not speak HTTP. TCP load balancing can be implemented at layer 4 or at layer 7. An HTTP load balancer is a reverse proxy that can perform extra actions on HTTPS traffic.

VPN

Virtual private networks, or VPNs, extend the reach of LANs without requiring owned or leased private lines. Companies can use VPNs to provide remote and mobile users with network access, connect geographically separated branches into a unified network and enable the remote use of applications that rely on internal servers.

VPNs can use one or both of two mechanisms. One is to use private circuits leased from a trusted communications provider: alone, this is called a *trusted VPN*. The other is to send encrypted traffic over the public Internet: alone, this is called a *secure VPN*. Using a secure VPN over a trusted VPN is called a hybrid VPN. Combining two kinds of secure VPN into one gateway, for instance, IPsec and Secure Sockets Layer (SSL), is also called a *hybrid VPN*.

Trusted VPNs

Over the years, implementations of trusted VPNs have moved from raw private circuits leased from telecommunications vendors to private IP network circuits leased from Internet providers. The major technologies used for implementing trusted VPNs over IP networks are ATM circuits, frame-relay circuits and Multiprotocol Label Switching (MPLS).

ATM and frame relay operate at the data link layer, which is Layer 2 of the OSI model. (Layer 1 is the physical layer; Layer 3 is the network layer.) MPLS emulates some properties of a circuit-switched network over a packet-switched network, and operates at a layer often referred to as "2.5" that is intermediate between the data link and the network. MPLS is beginning to replace ATM and frame relay to implement trusted VPNs for large corporations and service providers.

Secure VPNs

Secure VPNs can use IPsec with encryption, IPsec with Layer 2 Tunneling Protocol (L2TP), SSL 3.0 or Transport Layer Security (TLS) with encryption, Layer Two Forwarding (L2F) or Point-to-Point Tunneling Protocol (PPTP). [Editors' note: an earlier version of this article incorrectly stated that IPsec worked inside of L2TP, while the reverse is true]. Let's go over each of these briefly.

IPsec, or IP security, is a standard for encrypting and/or authenticating IP packets at the network layer. IPsec has a set of cryptographic protocols for two purposes: securing network packets and exchanging encryption keys. Some security experts, for instance, Bruce Schneier of Counterpane Internet Security Inc., have considered IPsec the preferred protocol for VPNs since the late 1990s. IPsec is supported in Windows XP, 2000, 2003 and Vista; in Linux 2.6 and later; in Mac OS X, NetBSD, FreeBSD and OpenBSD; in Solaris, AIX and HP-UX; and in VxWorks. Many vendors supply IPsec VPN servers and clients.

Microsoft has included PPTP clients in all versions of Windows since Windows 95 OSR2; PPTP clients are in Linux, Mac OS X, Palm PDA devices and Window Mobile 2003 devices. The company has also included PPTP servers in all its server products since Windows NT 4.0.

PPTP has been very popular, especially on Windows systems, because it is widely available, free and easy to set up. However, as implemented by Microsoft, it has not always been the most secure of the secure VPNs.

Schneier, with "Mudge" of L0pht Heavy Industries, found and published security flaws in Microsoft PPTP in 1998; Microsoft quickly fixed these issues with MS-CHAPv2 and MPPE, and Schneier and Mudge published an analysis confirming the improvements in 1999, but they pointed out that the security of Microsoft PPTP still depended on the security of each user's password. Microsoft has addressed this issue by enforcing password strength policies in its operating systems, but Schneier and Mudge still recommend IPsec rather than PPTP for secure VPNs as inherently safer.

L2TP combines ideas from PPTP and L2F, an older protocol developed by Cisco Systems Inc., to create a data link layer protocol. This provides a tunnel, but no security or authentication. L2TP can carry PPP sessions within its tunnel. Cisco implements L2TP in its routers. There are several open-source implementations of L2TP for Linux.

L2TP/IPsec combines L2TP's tunnel with IPsec's secure channel, which allows for easier secure Internet Key Exchange than pure IPsec. Microsoft has provided a free L2TP/IPsec VPN client for Windows 98, ME and NT since 2002, and ships an L2TP/IPsec VPN client with Windows XP, 2000, 2003 and Vista. Windows Server 2003 and Windows 2000 Server include L2TP/IPsec servers.

SSL and TLS are protocols for securing data flows at Layer 4 of the OSI model. SSL 3.0 and TLS 1.0, its successor, are commonly used with HTTP to enable secure Web browsing, called HTTPS. However, SSL/TLS can also be used to create a VPN tunnel. For example, OpenVPN is an open-source VPN package for Linux, xBSD, Mac OS X, Pocket PCs and Windows 2000, XP, 2003 and Vista, which uses SSL to provide encryption of both the data and control channels. Several vendors supply SSL VPN servers and clients.

Benefits and security risks of VPNs

A VPN can erase geographical barriers for a company, enable employees to work efficiently from home and allow a business to connect securely with its vendors and partners. A VPN is usually much cheaper to own and operate than private lines.

On the other hand, the use of a VPN can expose a company to potential security risks. While most VPNs in use are now fairly secure in and of themselves, a VPN can make it more difficult to secure the perimeter of a network properly. It is incumbent upon network administrators to apply the same security standards to computers connecting to the network via VPN as computers directly connected to the LAN.

Combining the use of two VPNs simultaneously can potentially expose one company's network to another's. In addition, using remote control software such as PC Anywhere, GoToMyPC or VNC in combination with a VPN can expose the company's network to the malware present on a remote computer that is not itself connection to the VPN.

Reliability, scalability and performance of VPNs

Because secure VPNs rely on encryption and some of the cryptographic functions used are computationally expensive, a heavily used VPN can load down its server. Administrators typically manage the server load by limiting the number of simultaneous connections to what the server can handle.

When the number of people attempting to connect to the VPN suddenly peaks, for example, during a storm that disrupts transportation, employees may find themselves unable to connect because all VPN ports are busy. That gives administrators motivation to make key applications

work without requiring the VPN, for instance, by setting up proxy servers or Internet Message Access Protocol servers to enable employees to access e-mail from home or from the road.

Deciding between IPsec and SSL/TLS for a given scenario can be complicated. One consideration is that SSL/TLS can work through a NAT-based firewall; IPsec cannot, but both protocols work through firewalls that do not translate addresses.

IPsec encrypts all IP traffic that flows between two computers. SSL/TLS is specific to an application. SSL/TLS uses expensive asymmetric encryption functions to establish a connection, and more efficient symmetric encryption functions to secure a running session.

In a real-world remote application, administrators may decide to mix and match protocols for the optimum balance of performance and security. For example, clients might connect to a Web-based front end through a firewall using a browser secured by SSL/TLS; the Web server might connect to an application server using IPsec; and the application server might connect to a database server across another firewall using SSL.

The scalability of VPNs can sometimes be improved by the use of dedicated server hardware. To cover that, however, we'd have to wade through the competing claims of VPN vendors: perhaps a subject for another day.

VPN resources

The Virtual Private Network Consortium maintains a list of its members, a table of IPsec VPN features supported by each vendor, and a table of SSL VPN features supported by each vendor. VPNC also supplies SimpleCA, a free, open-source certificate authority package for VPN administrators.

The Interop show runs in Las Vegas in the spring and New York in the fall, and usually attracts a wide range of VPN vendors and experts.

Types of VPN protocols

The above two VPN types are based on different VPN security protocols. Each of these VPN protocols offer different features and levels of security, and are explained below:

1. Internet Protocol Security or IPSec:

Internet Protocol Security or IPSec is used to secure Internet communication across an IP network. IPSec secures Internet Protocol communication by authenticating the session and encrypts each data packet during the connection.

IPSec operates in two modes, Transport mode and Tunneling mode, to protect data transfer between two different networks. The transport mode encrypts the message in the data packet and the tunneling mode encrypts the entire data packet. IPSec can also be used with other security protocols to enhance the security system.

2. Layer 2 Tunneling Protocol (L2TP):

L2TP or Layer 2 Tunneling Protocol is a tunneling protocol that is usually combined with another VPN security protocol like IPSec to create a highly secure VPN connection. L2TP creates a tunnel between two L2TP connection points and IPSec protocol encrypts the data and handles secure communication between the tunnel.

3. Point – to – Point Tunneling Protocol (PPTP):

PPTP or Point-to-Point Tunneling Protocol creates a tunnel and encapsulates the data packet. It uses a Point-to-Point Protocol (PPP) to encrypt the data between the connection. PPTP is one of the most widely used VPN protocol and has been in use since the time of Windows 95. Apart from Windows, PPTP is also supported on Mac and Linux.

4. Secure Sockets Layer (SSL) and Transport Layer Security (TLS):

SSL (Secure Sockets Layer) and TLS (Transport Layer Security) create a VPN connection where the web browser acts as the client and user access is restricted to specific applications instead of entire network. SSL and TLS protocol is most commonly used by online shopping websites and service providers. Web browsers switch to SSL with ease and with almost no action required from the user, since web browsers come integrated with SSL and TLS. SSL connections have https in the beginning of the URL instead of http.

5. OpenVPN:

OpenVPN is an open source VPN that is useful for creating Point-to-Point and Site-to-Site connections. It uses a custom security protocol based on SSL and TLS protocol.

6. Secure Shell (SSH):

Secure Shell or SSH creates the VPN tunnel through which the data transfer happens and also ensures that the tunnel is encrypted. SSH connections are created by a SSH client and data is transferred from a local port on to the remote server through the encrypted tunnel.

What is IPsec?

IPsec is a group of protocols that are used together to set up encrypted connections between devices. It helps keep data sent over public networks secure. IPsec is often used to set up VPNs, and it works by encrypting IP packets, along with authenticating the source where the packets come from.

Within the term "IPsec," "IP" stands for "Internet Protocol" and "sec" for "secure." The Internet Protocol is the main routing protocol used on the Internet; it designates where data will go using IP addresses. IPsec is secure because it adds encryption* and authentication to this process.

**Encryption is the process of concealing information by mathematically altering data so that it appears random. In simpler terms, encryption is the use of a "secret code" that only authorized parties can interpret.*

What is a VPN? What is an IPsec VPN?

A virtual private network (VPN) is an encrypted connection between two or more computers. VPN connections take place over public networks, but the data exchanged over the VPN is still private because it is encrypted.

VPNs make it possible to securely access and exchange confidential data over shared network infrastructure, such as the public Internet. For instance, when employees are working remotely instead of in the office, they often use VPNs to access corporate files and applications.

Many VPNs use the IPsec protocol suite to establish and run these encrypted connections. However, not all VPNs use IPsec. Another protocol for VPNs is SSL/TLS, which operates at a different layer in the OSI model than IPsec. (The OSI model is an abstract representation of the processes that make the Internet work.)

How do users connect to an IPsec VPN?

Users can access an IPsec VPN by logging into a VPN application, or "client." This typically requires the user to have installed the application on their device.

VPN logins are usually password-based. While data sent over a VPN is encrypted, if user passwords are compromised, attackers can log into the VPN and steal this encrypted data. Using two-factor authentication (2FA) can strengthen IPsec VPN security, since stealing a password alone will no longer give an attacker access.

How does IPsec work?

IPsec connections include the following steps:

Key exchange: Keys are necessary for encryption; a key is a string of random characters that can be used to "lock" (encrypt) and "unlock" (decrypt) messages. IPsec sets up keys with a key exchange between the connected devices, so that each device can decrypt the other device's messages.

Packet headers and trailers: All data that is sent over a network is broken down into smaller pieces called packets. Packets contain both a payload, or the actual data being sent, and headers, or information about that data so that computers receiving the packets know what to do with them. IPsec adds several headers to data packets containing authentication and encryption information. IPsec also adds trailers, which go after each packet's payload instead of before.

Authentication: IPsec provides authentication for each packet, like a stamp of authenticity on a collectible item. This ensures that packets are from a trusted source and not an attacker.

Encryption: IPsec encrypts the payloads within each packet and each packet's IP header (unless transport mode is used instead of tunnel mode — see below). This keeps data sent over IPsec secure and private.

Transmission: Encrypted IPsec packets travel across one or more networks to their destination using a transport protocol. At this stage, IPsec traffic differs from regular IP traffic in that it most

often uses UDP as its transport protocol, rather than TCP. TCP, the Transmission Control Protocol, sets up dedicated connections between devices and ensures that all packets arrive. UDP, the User Datagram Protocol, does not set up these dedicated connections. IPsec uses UDP because this allows IPsec packets to get through firewalls.

Decryption: At the other end of the communication, the packets are decrypted, and applications (e.g. a browser) can now use the delivered data.

What protocols are used in IPsec?

In networking, a protocol is a specified way of formatting data so that any networked computer can interpret the data. IPsec is not one protocol, but a suite of protocols. The following protocols make up the IPsec suite:

Authentication Header (AH): The AH protocol ensures that data packets are from a trusted source and that the data has not been tampered with, like a tamper-proof seal on a consumer product. These headers do not provide any encryption; they do not help conceal the data from attackers.

Encapsulating Security Protocol (ESP): ESP encrypts the IP header and the payload for each packet — unless transport mode is used, in which case it only encrypts the payload. ESP adds its own header and a trailer to each data packet.

Security Association (SA): SA refers to a number of protocols used for negotiating encryption keys and algorithms. One of the most common SA protocols is Internet Key Exchange (IKE).

Finally, while the **Internet Protocol (IP)** is not part of the IPsec suite, IPsec runs directly on top of IP.

What is the difference between IPsec tunnel mode and IPsec transport mode?

IPsec tunnel mode is used between two dedicated routers, with each router acting as one end of a virtual "tunnel" through a public network. In IPsec tunnel mode, the original IP header containing the final destination of the packet is encrypted, in addition to the packet payload. To

tell intermediary routers where to forward the packets, IPsec adds a new IP header. At each end of the tunnel, the routers decrypt the IP headers to deliver the packets to their destinations.

In transport mode, the payload of each packet is encrypted, but the original IP header is not. Intermediary routers are thus able to view the final destination of each packet — unless a separate tunneling protocol (such as GRE) is used.

What port does IPsec use?

A network port is the virtual location where data goes in a computer. Ports are how computers keep track of different processes and connections; if data goes to a certain port, the computer's operating system knows which process it belongs to. IPsec usually uses port 500.

How does IPsec impact MSS and MTU?

MSS and MTU are two measurements of packet size. Packets can only reach a certain size (measured in bytes) before computers, routers, and switches cannot handle them. MSS measures the size of each packet's payload, while MTU measures the entire packet, including headers. Packets that exceed a network's MTU may be fragmented, meaning broken up into smaller packets and then reassembled. Packets that exceed the MSS are simply dropped.

IPsec protocols add several headers and trailers to packets, all of which take up several bytes. For networks that use IPsec, either the MSS and MTU have to be adjusted accordingly, or packets will be fragmented and slightly delayed. Usually, the MTU for a network is 1,500 bytes. A normal IP header is 20 bytes long, and a TCP header is also 20 bytes long, meaning each packet can contain 1,460 bytes of payload. However, IPsec adds an Authentication Header, an ESP header, and associated trailers. These add 50-60 bytes to a packet, or more.

Understanding IPSec Modes –Tunnel Mode & Transport Mode

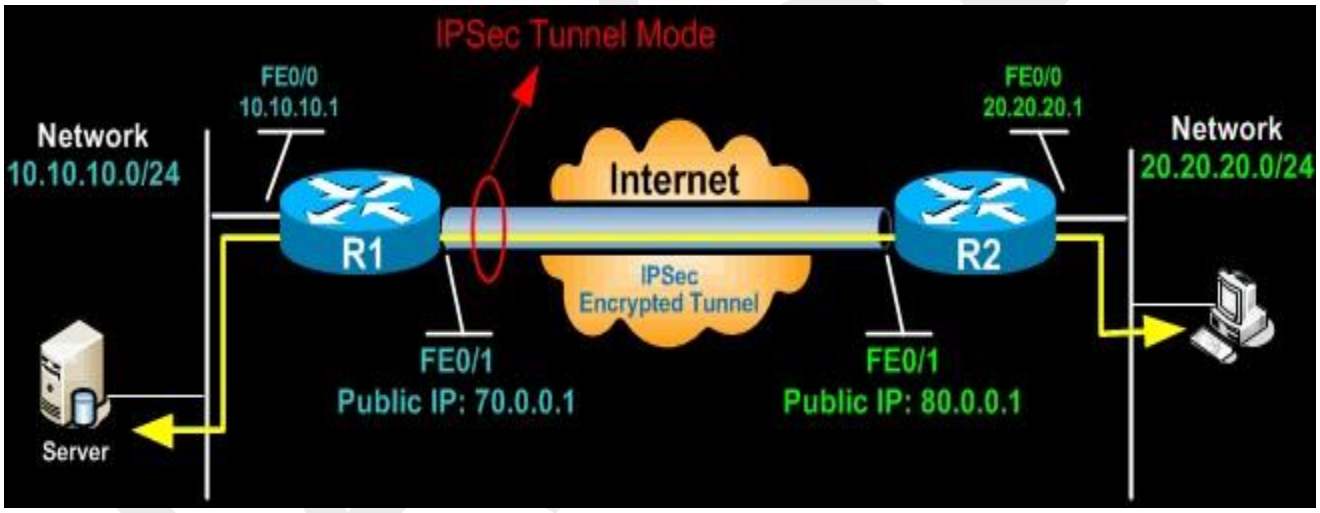
IPSec can be configured to operate in two different modes, Tunnel and Transport mode. Use of each mode depends on the requirements and implementation of IPSec.

IPSec Tunnel Mode

IPSec tunnel mode is the **default mode**. With tunnel mode, the entire original IP packet is protected by IPSec. This means IPSec wraps the original packet, encrypts it, adds a new IP header and sends it to the other side of the VPN tunnel (IPSec peer).

Tunnel mode is most commonly used between gateways (Cisco routers or ASA firewalls), or at an end-station to a gateway, the gateway acting as a proxy for the hosts behind it.

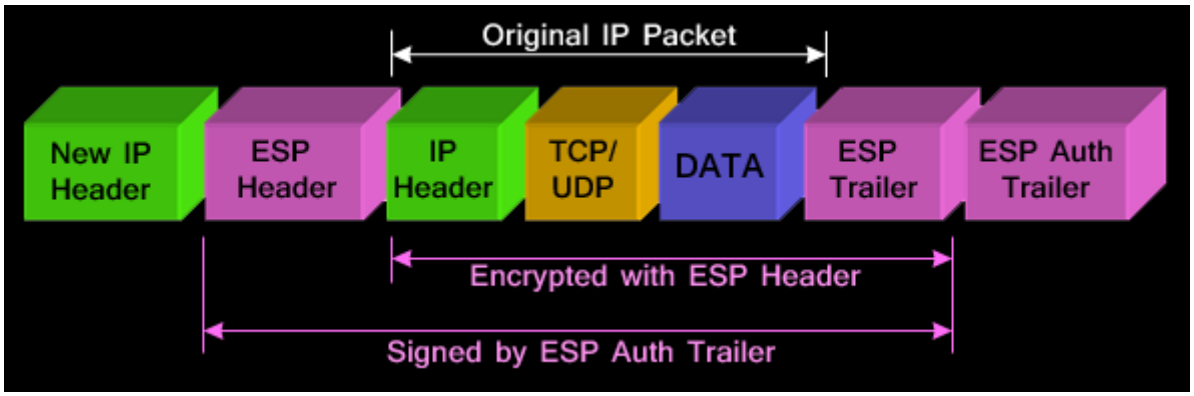
Tunnel mode is used to encrypt traffic between secure IPSec Gateways, for example two Cisco routers connected over the Internet via IPSec VPN. In this example, each router acts as an IPSec Gateway for their LAN, providing secure connectivity to the remote network:



Another example of tunnel mode is an IPSec tunnel between a Cisco VPN Client and an IPSec Gateway (e.g ASA5510 or PIX Firewall). The client connects to the IPSec Gateway. Traffic from the client is encrypted, encapsulated inside a new IP packet and sent to the other end. Once decrypted by the firewall appliance, the client's original IP packet is sent to the local network.

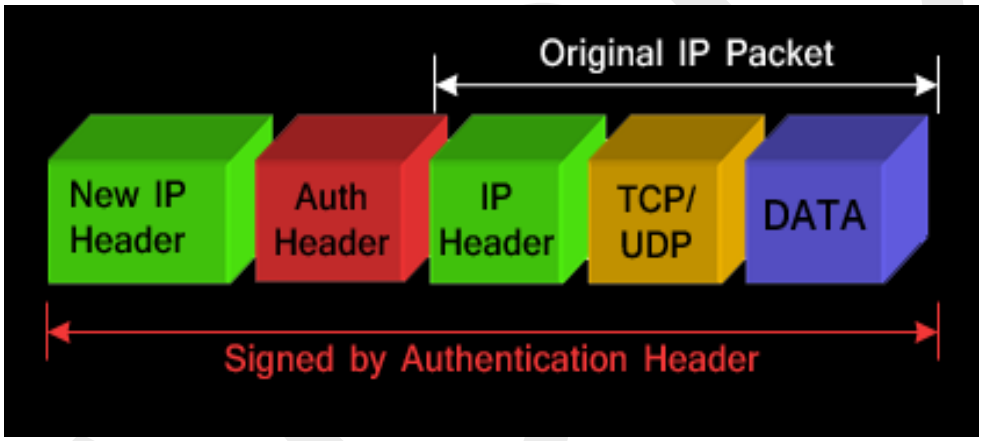
In tunnel mode, an IPSec header (**AH** or **ESP header**) is inserted between the IP header and the upper layer protocol. Between AH and ESP, ESP is most commonly used in IPSec VPN Tunnel configuration.

The packet diagram below illustrates **IPSec Tunnel mode** with **ESP header**:



ESP is identified in the **New IP header** with an IP **protocol ID** of 50.

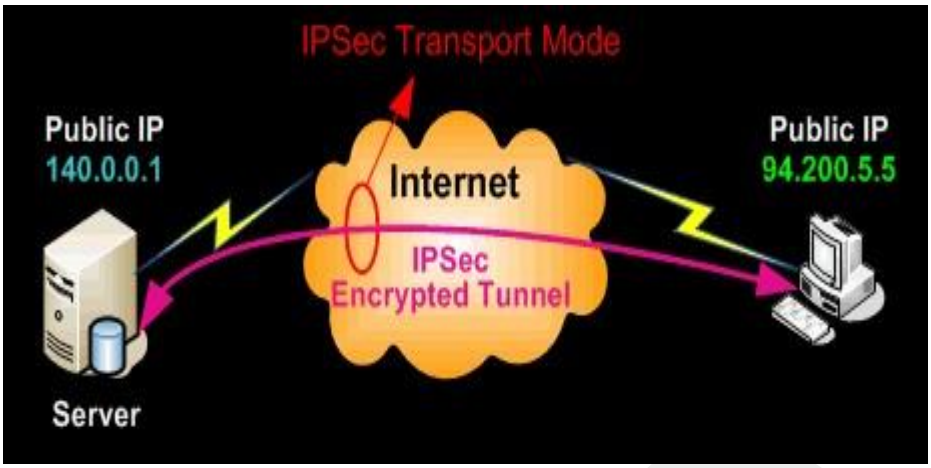
The packet diagram below illustrates **IPSec Tunnel mode** with **AH header**:



The AH can be applied alone or together with the ESP, when IPSec is in tunnel mode. AH's job is to protect the entire packet. The AH does not protect all of the fields in the New IP Header because some change in transit, and the sender cannot predict how they might change. The AH protects everything that does not change in transit. AH is identified in the **New IP header** with an IP **protocol ID** of **51.**

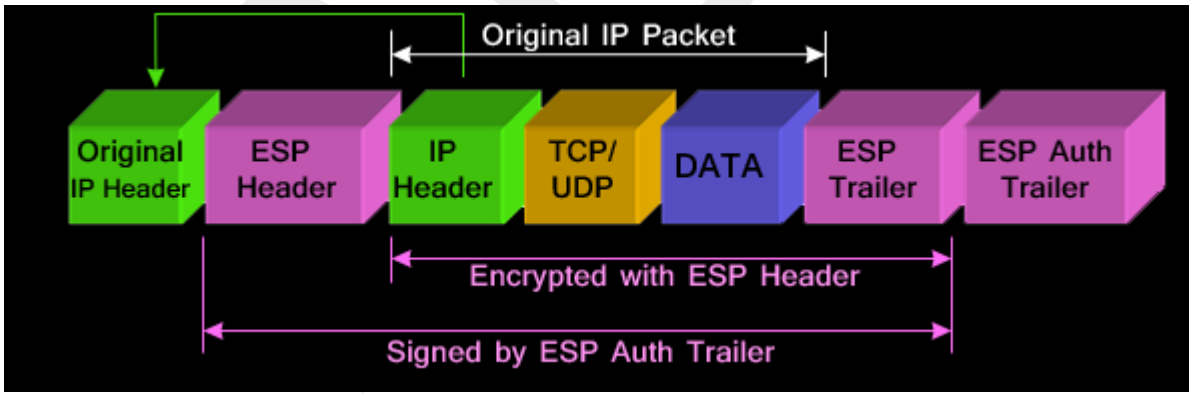
IPSec Transport Mode

IPSec Transport mode is used for end-to-end communications, for example, for communication between a client and a server or between a workstation and a gateway (if the gateway is being treated as a host). A good example would be an encrypted Telnet or Remote Desktop session from a workstation to a server.



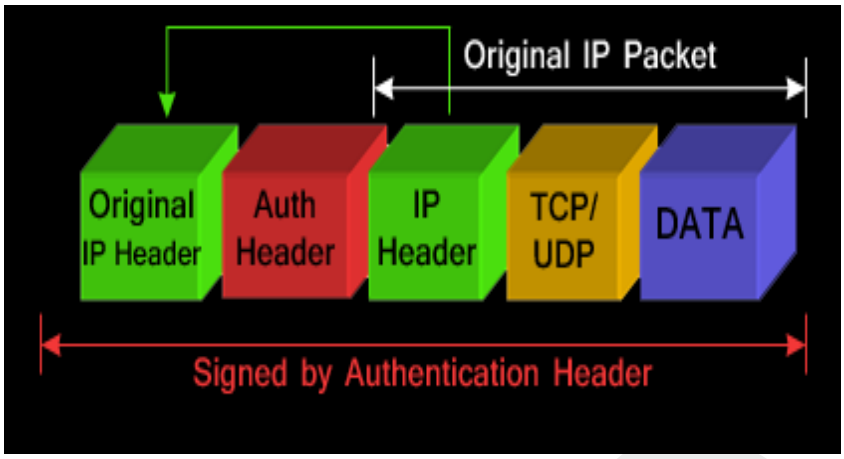
Transport mode provides the protection of our data, also known as IP Payload, and consists of TCP/UDP header + Data, through an AH or ESP header. The payload is encapsulated by the IPSec headers and trailers. The original IP headers remain intact, except that the IP protocol field is changed to ESP (50) or AH (51), and the original protocol value is saved in the IPsec trailer to be restored when the packet is decrypted.

IPSec transport mode is usually used when another tunneling protocol (like GRE) is used to first encapsulate the IP data packet, then IPSec is used to protect the GRE tunnel packets. IPSec protects the GRE tunnel traffic in transport mode.
The packet diagram below illustrates **IPSec Transport mode** with **ESP header**:



Notice that the original IP Header is **moved** to the front. Placing the sender's IP header at the front (with minor changes to the protocol ID), proves that transport mode does not provide protection or encryption to the original IP header and ESP is identified in the **New IP header** with an IP **protocol ID** of **50**.

The packet diagram below illustrates **IPSec Transport mode** with **AH header**:



The AH can be applied alone or together with the ESP when IPSec is in transport mode. AH's job is to **protect** the entire packet, however, IPSec in transport mode does not create a new IP header in front of the packet but places a copy of the original with some minor changes to the protocol ID therefore not providing essential protection to the details contained in the IP header (Source IP, destination IP etc). AH is identified in the **New IP header** with an IP **protocol ID** of **51.**

VPN Connection Types

OSU has two types of VPN connections available:

- **Full Tunnel (Default) -** Routes and encrypts ALL requests through the VPN to OSU, regardless of where the service is hosted. Note that when connected via full tunnel, it is not possible to access local network resources. Full tunnel is generally recommended because it is more secure.
- **Split Tunnel** - Routes and encrypts all OSU-bound requests over the VPN. Traffic destined to sites on the Internet (including Zoom, Canvas, Office 365, and Google) does not go through the VPN server in split tunnel mode.

Use Split Tunnel or Full Tunnel?

Full tunnel is the better option if you handle confidential data. It is also better to use full tunnel if you are connecting from a network you do not trust, such as in a coffee shop or hotel.

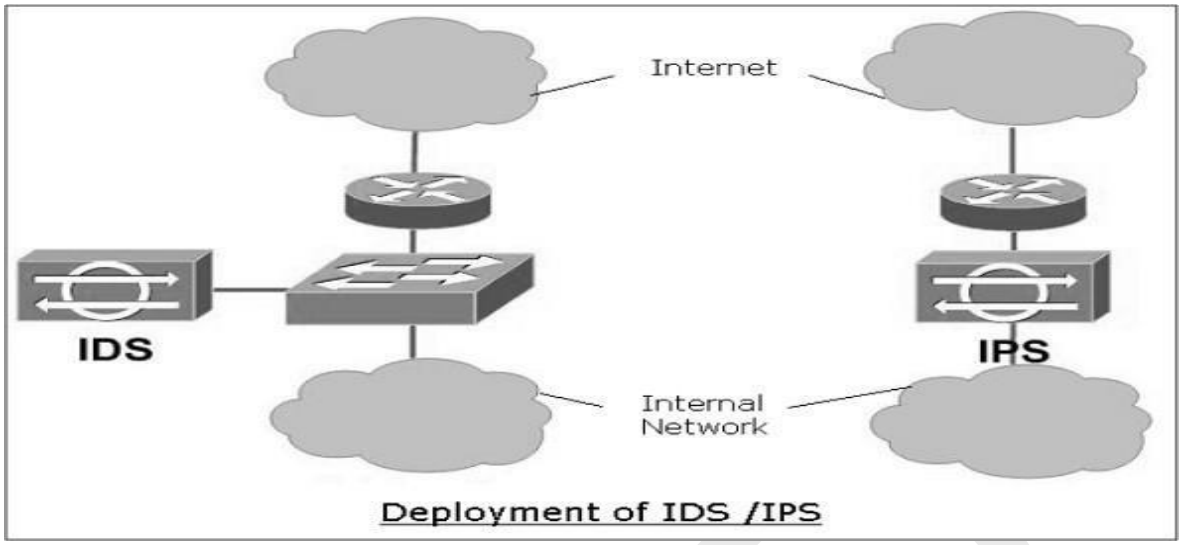
Split tunnel may be necessary if you need to access both local resources and OSU resources at the same time, but it is less secure.

Intrusion Detection / Prevention System

- The packet filtering firewalls operate based on rules involving TCP/UDP/IP headers only. They do not attempt to establish correlation checks among different sessions.
- Intrusion Detection/Prevention System (IDS/IPS) carry out Deep Packet Inspection (DPI) by looking at the packet contents. For example, checking character strings in packet against database of known virus, attack strings.
- Application gateways do look at the packet contents but only for specific applications.
- They do not look for suspicious data in the packet.
- IDS/IPS looks for suspicious data contained in packets and tries to examine correlation among multiple packets to identify any attacks such as port scanning, network mapping, and denial of service and so on.

Difference between IDS and IPS

- IDS and IPS are similar in detection of anomalies in the network.
- IDS is a ‘visibility’ tool whereas IPS is considered as a ‘control’ tool.
- Intrusion Detection Systems sit off to the side of the network, monitoring traffic at many different points, and provide visibility into the security state of the network.
- In case of reporting of anomaly by IDS, the corrective actions are initiated by the network administrator or other device on the network.
- Intrusion Prevention System are like firewall and they sit in-line between two networks and control the traffic going through them.
- It enforces a specified policy on detection of anomaly in the network traffic. Generally, it drops all packets and blocks the entire network traffic on noticing an anomaly till such time an anomaly is addressed by the administrator.



Deployment of IDS /IPS

Types of IDS

There are two basic types of IDS.

- **Signature-based IDS**
  - It needs a database of known attacks with their signatures.
  - Signature is defined by types and order of packets characterizing a particular attack.
  - Limitation of this type of IDS is that only known attacks can be detected. This IDS can also throw up a false alarm. False alarm can occur when a normal packet stream matches the signature of an attack.
  - Well-known public open-source IDS example is “Snort” IDS.
- **Anomaly-based IDS**
  - This type of IDS creates a traffic pattern of normal network operation.
  - During IDS mode, it looks at traffic patterns that are statistically unusual. For example, ICMP unusual load, exponential growth in port scans, etc.
  - Detection of any unusual traffic pattern generates the alarm.
  - The major challenge faced in this type of IDS deployment is the difficulty in distinguishing between normal traffic and unusual traffic.

What Is a Network Attack?

A network attack is an attempt to gain unauthorized access to an organization's network, with the objective of stealing data or perform other malicious activity. There are two main types of network attacks:

- **Passive:** Attackers gain access to a network and can monitor or steal sensitive information, but without making any change to the data, leaving it intact.
- **Active:** Attackers not only gain unauthorized access but also modify data, either deleting, encrypting or otherwise harming it.

We distinguish network attacks from several other types of attacks:

- **Endpoint attacks**—gaining unauthorized access to user devices, servers or other endpoints, typically compromising them by infecting them with malware.
- **Malware attacks**—infecting IT resources with malware, allowing attackers to compromise systems, steal data and do damage. These also include ransomware attacks.
- **Vulnerabilities, exploits and attacks**—exploiting vulnerabilities in software used in the organization, to gain unauthorized access, compromise or sabotage systems.
- **Advanced persistent threats**—these are complex multilayered threats, which include network attacks but also other attack types.

In a network attacks, attackers are focused on penetrating the corporate network perimeter and gaining access to internal systems. Very often, once inside attackers will combine other types of attacks, for example compromising an endpoint, spreading malware or exploiting a vulnerability in a system within the network.

What are the Common Types of Network Attacks?

Following are common threat vectors attackers can use to penetrate your network.

1. Unauthorized access

Unauthorized access refers to attackers accessing a network without receiving permission.

Among the causes of unauthorized access attacks are weak passwords, lacking protection against social engineering, previously compromised accounts, and insider threats.

2. Distributed Denial of Service (DDoS) attacks

Attackers build botnets, large fleets of compromised devices, and use them to direct false traffic at your network or servers. DDoS can occur at the network level, for example by sending huge volumes of SYN/ACC packets which can overwhelm a server, or at the application level, for example by performing complex SQL queries that bring a database to its knees.

3. Man in the middle attacks

A man in the middle attack involves attackers intercepting traffic, either between your network and external sites or within your network. If communication protocols are not secured or attackers find a way to circumvent that security, they can steal data that is being transmitted, obtain user credentials and hijack their sessions.

4. Code and SQL injection attacks

Many websites accept user inputs and fail to validate and sanitize those inputs. Attackers can then fill out a form or make an API call, passing malicious code instead of the expected data values. The code is executed on the server and allows attackers to compromise it.

5. Privilege escalation

Once attackers penetrate your network, they can use privilege escalation to expand their reach. Horizontal privilege escalation involves attackers gaining access to additional, adjacent systems, and vertical escalation means attackers gain a higher level of privileges for the same systems.

6. Insider threats

A network is especially vulnerable to malicious insiders, who already have privileged access to organizational systems. Insider threats can be difficult to detect and protect against, because insiders do not need to penetrate the network in order to do harm. New technologies like User and Even Behavioral Analytics (UEBA) can help identify suspicious or anomalous behavior by internal users, which can help identify insider attacks.

Intrusion Detection Systems (IDS)

An intrusion detection system (IDS) is a system that monitors network traffic to detect abnormal behavior and content. In case of a successful detection, an alarm is raised or a procedure to prevent the attack is taken. A system that takes appropriate actions upon a detection is also known as Intrusion Prevention System (IPS).The outcome with respect to the correctness of the decision made by the intrusion detection system, can be classified into four categories described in table

Output	Description
True Positive (TP)	Identifies an activity as an intrusion and the activity is actually an intrusion
True Negative (TN)	Identifies system behavior as normal and the activity is actually normal
False Positive (FP)	Identifies an activity as an intrusion but the activity is normal
False Negative (FN)	Identifies an activity as normal when the activity is an intrusion

False negative cases are considered to be the problematic, because of theirgiven nature of not detecting an actual intrusion on the network, which can lead toa variety of unforeseen problems. When designing an IDS, it is desired to diminishthe false positive rate to a minimum, in order to not flood the logs with erroneousresults. Therefore, when evaluating an intrusion detection system it is recommendedto aim for the true-positive and true-negative rate to be as high as possible, as wellas the false positive and false negative to be as low as possible.

Intrusion detection system types and classification

This gives a classification on the different types of IDS and how they perform.Figure exhibits the major classification of IDS.

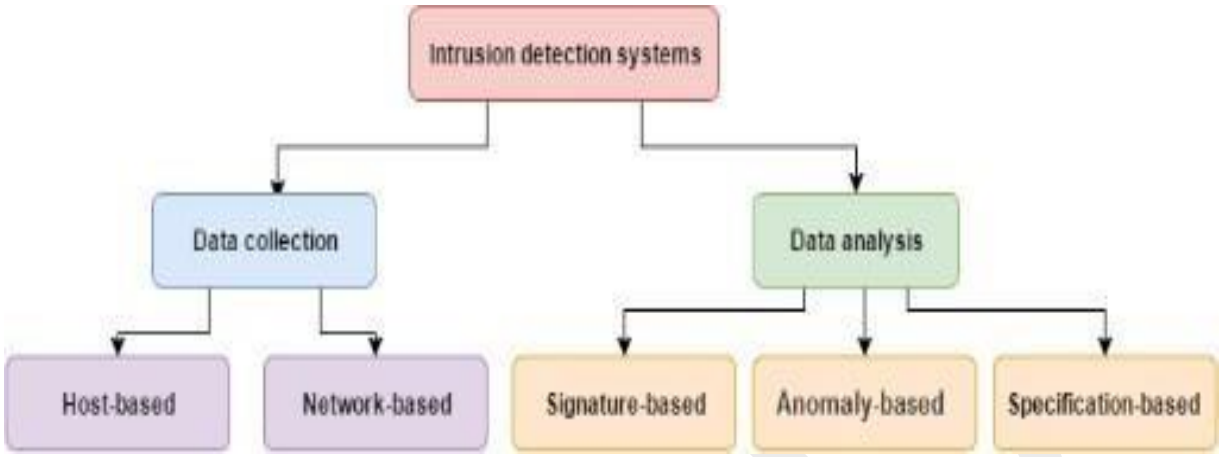


Figure : Intrusion detection systems classification

The first step of designing an IDS, is to think about the placement of the IDSand which data needed to be collected. The IDS can be host-based, which wouldcollect information about the file system and the behavioural patterns of the user ofthe system. The second approach features a network-based approach, in which themain goal is to monitor the traffic that is entering and leaving the network.

A signature-based approach is simply looking for signatures of known attackvectors and tries to find them in the in collected traffic. The anomaly-based ap-proach relies on a "baseline" condition of the system and reports everything that isstriving away from this baseline. The specification-based approach is a relativelynew attempt to somehow connect the aforementioned approaches. The main ideais to define a "legal behaviour" of the communication which is following a certainprotocol.

Data collection techniques

Intrusion detection systems can be classified according to the type of data they col-lect. There are two main categories for these types of IDSs: Network-based IntrusionDetection System (NIDS) and Host-based Intrusion Detection System (HIDS). Thenetwork-based IDS, monitors the network traffic and it is usually placed on thetraffic routing component, such as a network gateway, which is connecting multiplenetworks together. This enables the NIDS to intercept and analyze the traffic beforeit is entering a sub-network, e.g the local area network (LAN)

A host-based IDS, contrary to network-based, is placed on the host computeritself where it monitors the host's system behaviour, e.g. which processes are ac-cessing which resources. In addition to these main classifications of IDS, othersub-categories exist [6], such as:

•**Stack-based**: monitors the exchange of data between different layers of theprotocol's stack.

•**Protocol based**: Monitors the protocols that are used by the host system

•**Graph-based**: monitors connections between several nodes or hosts

Data analysis techniques

As mentioned in the introduction to this chapter, there are three main types ofintrusion detection approaches: Anomaly-based detection, signature-based detectionand specification-based detection. In this section, these different paradigms areexplained in greater detail.

Signature-based detection

Signature-based detection observes the messages and tries to find pre-defined pat-terns or sequences. These patterns are also known as signatures.

The Signatures can mainly define two kinds of lists, a white-list or a blacklist. In the white list, we only define the kind of messages that are allowed in thesystem and in the blacklist we define the types of patterns and messages that arenot allowed to access the system.

Both have their advantages and disadvantages. When using white-listing ,we need to to have a prior knowledge of the exchanged messages. This might havehuge constraints on open systems , for example computers that use application layerprotocols that have very diverse contents. On the other hand, this would be thebest choice closed systems and for the lower level protocols that have very limitedcapabilities and fewer types of messages with predictable content. Correspondingly,blacklisting is suitable for more complex systems. However, the challenge withblacklisting is the need to define the patterns and messages that are not allowed inthe system

Anomaly-based detection

In anomaly detection, we observe the behaviour of the system. For this, we need to define an abnormal attitude that will be considered as suspicious. This is similar to the blacklisting approach in the signature based, however, in anomaly based wedefine a behaviour that can have a much broader scope and result in detecting behaviour that has never been seen before.

In order to model the behaviour of the host system or the network traffic ob-served, there is a need to construct a view which has a certain level of abstraction, that leads to extraction of features. These features are then used together with ma-chine learning techniques that can construct a hypothesis that models the behaviour of the system e.g Neural networks, support vector machines, time series , standard deviation and more. Upon the IDS run-time, features similar to the one used to train the model, have to be extracted again before passing them to the hypothesis that will perform the classification.

Specification-based detection

In specification based intrusion detection system, a set of properties, that are ex-tracted from the protocol design are defined for the monitoring purpose i.e. we know the correct system behavior and can specify it in rules. The classification and detection is then performed by observing a deviation of the execution from the defined properties.

Stateful vs Stateless Intrusion detection systems

Another general sub-classification for intrusion detection systems are: stateless and stateful. A stateless intrusion detection system is a system that does not keep track of previous data it has already seen. While stateful intrusion detection systems, keep temporary information about previously seen data in order to use it for possible further detections.

Intrusion detection system architecture

The following section focuses on the basic system architecture of a network-based intrusion detection system. This will aid in gaining a deeper understanding of how to plan and build the prototype IDS .

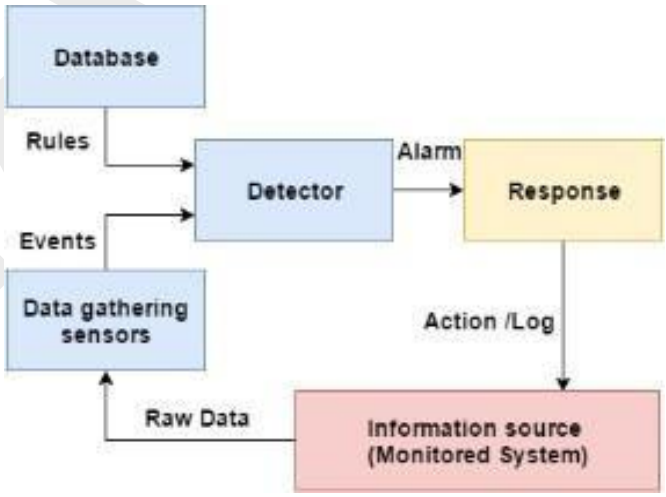
When implementing an Intrusion detection system, it is not possible to completely resort to a common standard due to varying requirements when utilizing the gathering of the data and its analysis. Nevertheless, almost all of them have some basic components/modules that they all share . These components are:

•**Data gathering**: used for monitoring the source environment. The data gathering is performed using different sensors that observe a specific application or protocol. A pre-processing module can also be included [3] that performs basic classification of the data type received from the source.

•**Detector (Detection engine**): is a module that performs the comparison between the gathered data and the defined rules set and raises alarms in case a deviation is found.

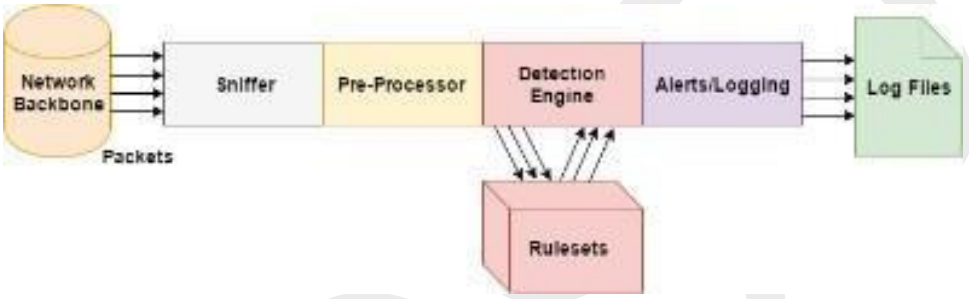
•**Database(Knowledgebase)**: is a storage module that contains the rule-sets or the IDs which the detector uses when comparing the received data.

•**Output (Response):** When an alarm is raised a proper action is taken. This could be an active response where the IDS performs a predefined action such as drop the packet, or an inactive response such as logging for later inspection by a human factor to determine the appropriate response



Intrusion detection system architecture

We use Snort, an IDS/IPS [3] to explain how a generic IDS works. Figure shows the architecture of Snort. We can see that the packet arrives from the network where it is collected by the sniffer module. The packet is then sent tothe pre-processor that inspects the type of the packet the detection engine will deal with. This information together with the packet are then forwarded to the detection engine that compares the internal component of this packet with the predefined rule set stored. Based on the decision of the detection engine an alert is raised and logged in the log database.



Snort IDS architecture

Dos and Ddos

A **DoS attack** is a denial of service attack where a computer is used to flood a server with TCP and UDP packets. A **DDoS attack** is where multiple systems target a single system with a DoS attack. The targeted network is then bombarded with packets from multiple locations.

Denial of Service (DoS) and **Distributed Denial of Service (DDoS)** attacks are two of the most intimidating threats that modern enterprises face. Few forms of attack can have the financial ramifications as that of a successful DoS attack. Security surveys indicate that the cost of a DDoS attack averages between $20,000-$40,000 per hour. This is an astronomical figure and can put even the largest organizations under pressure.

What is a DoS Attack?

During this type of attack, the service is put out of action as the packets sent over the network to **overload the server's capabilities and make the server unavailable** to other devices and users throughout the network. DoS attacks are used to shut down individual machines and networks so that they can't be used by other users.

There are a number of different ways that DoS attacks can be used. These include the following:

- **Buffer overflow attacks** – This type of attack is the most common DOS attack experienced. Under this attack, the attacker overloads a network address with traffic so that it is put out of use.
- **Ping of Death or ICMP flood** – An ICMP flood attack is used to take unconfigured or misconfigured network devices and uses them to send spoof packets to ping every computer within the target network. This is also known as a ping of death (POD) attack.
- **SYN flood** – SYN flood attacks send requests to connect to a server but don't complete the handshake. The end result is that the network becomes inundated with connection requests that prevent anyone from connecting to the network.
- **Teardrop Attack** – During a teardrop DoS attack, an attacker sends IP data packet fragments to a network. The network then attempts to recompile these fragments into their original packets. The process of compiling these fragments exhausts the system

and it ends up crashing. It crashes because the fields are designed to confuse the system so that it can not put them back together.

The ease with which DoS attacks can be coordinated has meant that they have become **one of the most pervasive cybersecurity threats** that modern organizations have to face. DoS attacks are simple but effective and can bring about devastating damage to the companies or individuals they are aimed at. With one attack, an organization can be put out of action for days or even weeks.

The time an organization spends offline adds up. Being unable to access the network costs organizations thousands every year. Data may not be lost but the disruption to service and downtime can be massive. Preventing DoS attacks is one of the basic requirements of staying protected in the modern age.

What is a DDoS Attack?

The reason for this is that there is a larger number of machines at the attackers' disposal and it becomes difficult for the victim to pinpoint the origin of the attack.

In addition, using a DDoS attack **makes it more complicated for the victim to recover**. Nine times out of ten the systems used to execute DDoS attacks have been compromised so that the attacker can launch attacks remotely through the use of slave computers. These slave computers are referred to as zombies or bots.

These bots form a network of connected devices called a botnet that is managed by the attacker through a command and control server. The command and control server allows the attacker or botmaster to coordinate attacks. Botnets can be made up of anywhere between a handful of bots to hundreds of different bots.

Broad Types of DoS and DDoS Attacks

There are a number of broad categories that DoS attacks fall into for taking networks offline. These come in the form of:

- **Volumetric Attacks –** Volumetric attacks are classified as any form of attack where a target network's bandwidth resources are deliberately consumed by an attacker. Once network bandwidth has been consumed it is unavailable to legitimate devices and users

within the network. Volumetric attacks occur when the attacker floods network devices with ICMP echo requests until there is no more bandwidth available.

- **Fragmentation Attacks –** Fragmentation attacks are any kind of attack that forces a network to reassemble manipulated network packets. During a fragmentation attack the attacker sends manipulated packets to a network so that once the network tries to reassemble them, they can't be reassembled. This is because the packets have more packet header information than is permitted. The end result is packet headers which are too large to reassemble in bulk.
- **TCP-State Exhaustion Attacks –** In a TCP-State Exhaustion attack the attacker targets a web server or firewall in an attempt to limit the number of connections that they can make. The idea behind this style of attack is to push the device to the limit of the number of concurrent connections.
- **Application Layer Attacks –** Application layer or Layer 7 attacks are attacks that target applications or servers in an attempt to use up resources by creating as many processes and transactions possible. Application layer attacks are particularly difficult to detect and address because they don't need many machines to launch an attack.

Most Common Forms of DDoS Attacks

As you can see, DDoS attacks are the more complex of the two threats because they use a range of devices that increase the severity of attacks. Being attacked by one computer is not the same as being attacked by a botnet of one hundred devices!

Part of being prepared for DDoS attacks is being familiar with as many different attack forms as you can. In this section, we're going to look at these in further detail so you can see how these attacks are used to damage enterprise networks.

DDoS attacks can come in various forms including:

- **Ping of Death –** During a Ping of Death (POD) attack the attacker sends multiple pings to one computer. POD attacks use manipulated packets to send packets to the network which have IP packets that are larger than the maximum packet length. These illegitimate packets are sent as fragments. Once the victim's network attempts to reassemble these packets network resources are used up, they are unavailable to legitimate packets. This grinds the target network to a halt and takes it out of action completely.

- **UDP Floods –** A UDP flood is a DDoS attack that floods the victim network with User Datagram Protocol (UDP) packets. The attack works by flooding ports on a remote host so that the host keeps looking for an application listening at the port. When the host discovers that there is no application it replies with a packet that says the destination wasn't reachable. This consumes network resources and means that other devices can't connect properly.
- **Ping Flood –** Much like a UDP flood attack, a ping flood attack uses ICMP Echo Request or ping packets to derail a network's service. The attacker sends these packets rapidly without waiting for a reply in an attempt to make the target network unreachable through brute force. These attacks are particularly concerning because bandwidth is consumed both ways with attacked servers trying to reply with their own ICMP Echo Reply packets. The end result is a decline in speed across the entire network.
- **SYN Flood –** SYN Flood attacks are another type of DoS attack where the attacker uses the TCP connection sequence to make the victim's network unavailable. The attacker sends SYN requests to the victim's network which then responds with a SYN-ACK response. The sender is then supposed to respond with an ACK response but instead, the attacker doesn't respond (or uses a spoofed source IP address to send SYN requests instead). Every request that goes unanswered takes up network resources until no devices can make a connection.
- **Slowloris –** Slowloris is a type of DDoS attack software that was originally developed by Robert Hansen or RSnake to take down web servers. A Slowloris attack occurs when the attacker sends partial HTTP requests with no intention of completing them. To keep the attack going, Slowloris periodically sends HTTP headers for each request to keep the computer network's resources tied up. This continues until the server can't make any more connections. This form of attack is used by attackers because it doesn't require any bandwidth.
- **HTTP Flood –** In a HTTP Flood attack the attacker users HTTP GET or POST requests to launch an assault on an individual web server or application. HTTP floods are a Layer 7 attack and don't use malformed or spoofed packets. Attackers use this type of attack because they require less bandwidth than other attacks to take the victim's network out of operation.

- **Zero-Day Attacks –** Zero-Day attacks are attacks that exploit vulnerabilities that have yet to be discovered. This is a blanket term for attacks that could be faced in the future. These types of attacks can be particularly devastating because the victim has no specific way to prepare for them before experiencing a live attack
- **Difference between DOS and DDOS attack :**

DOS	DDOS
DOS Stands for Denial of service attack.	DDOS Stands for Distributed Denial of service attack.
In Dos attack single system targets the victims system.	In DDos multiple system attacks the victims system..
Victim PC is loaded from the packet of data sent from a single location.	Victim PC is loaded from the packet of data sent from Multiple location.
Dos attack is slower as compared to ddos.	DDos attack is faster than Dos Attack.
Can be blocked easily as only one system is used.	It is difficult to block this attack as multiple devices are sending packets and attacking from multiple locations.
In DOS Attack only single device is used with DOS Attack tools.	In DDos attack Bots are used to attack at the same time.
DOS Attcaks are Easy to trace.	DDOS Attacks are Difficult to trace.
Volume of traffic in Dos attack is less as compared to DDos.	DDoS attacks allow the attacker to send massive volumes of traffic to the victim network.
Types of DOS Attacks are: 1. Buffer overflow attacks 2. Ping of Death or ICMP flood 3. Teardrop Attack	Types of DDOS Attacks are: 1. Volumetric Attacks 2. Fragmentation Attacks 3. Application Layer Attacks

Defense-in-depth

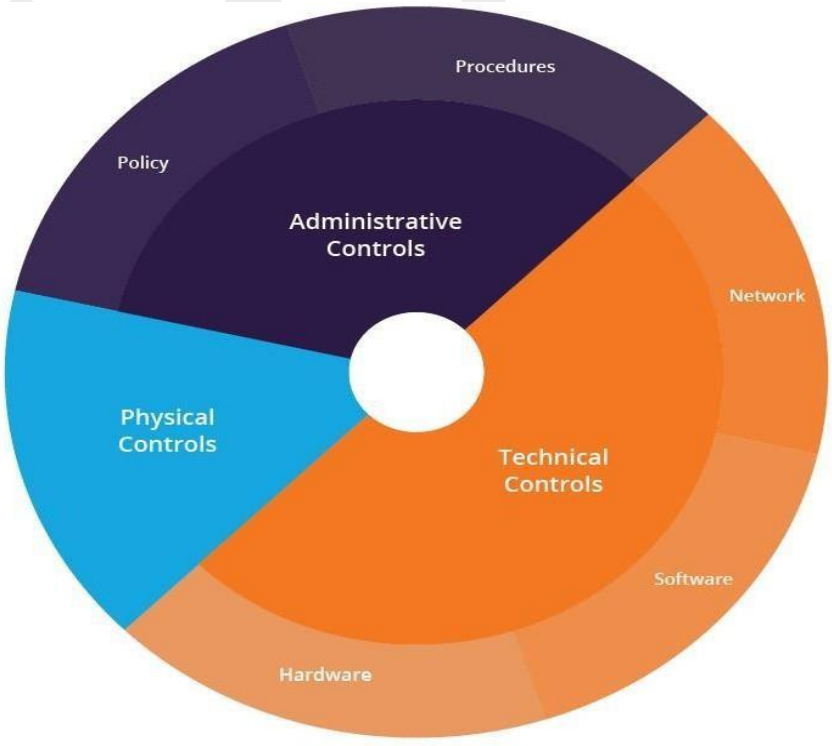
Defense-in-depth is an information assurance strategy that provides multiple, redundant defensive measures in case a security control fails or a vulnerability is exploited. It originates from a military strategy by the same name, which seeks to delay the advance of an attack, rather than defeating it with one strong line of defense.

Defense-in-depth cybersecurity use cases include end-user security, product design and network security.

An opposing principle to defense in depth is known as simplicity-in-security, which operates under the assumption that too many security measures might introduce problems or gaps that attackers can leverage.

Defense-in-depth architecture: Layered security

Defense-in-depth security architecture is based on controls that are designed to protect the physical, technical and administrative aspects of your network.



Defense in depth, layered security architecture

- **Physical controls** – These controls include security measures that prevent physical access to IT systems, such as security guards or locked doors.
- **Technical controls** – Technical controls include security measures that protect network systems or resources using specialized hardware or software, such as a firewall appliance or antivirus program.
- **Administrative controls** – Administrative controls are security measures consisting of policies or procedures directed at an organization's employees, e.g., instructing users to label sensitive information as "confidential".

Additionally, the following security layers help protect individual facets of your network:

- **Access measures** – Access measures include authentication controls, biometrics, timed access and VPN.
- **Workstation defenses** – Workstation defense measures include antivirus and anti-spam software.
- **Data protection** – Data protection methods include data at rest encryption, hashing, secure data transmission and encrypted backups.
- **Perimeter defenses** – Network perimeter defenses include firewalls, intrusion detection systems and intrusion prevention systems.
- **Monitoring and prevention** – The monitoring and prevention of network attacks involves logging and auditing network activity, vulnerability scanners, sandboxing and security awareness training.

Defense-in-depth information assurance: Use cases

Broadly speaking, defense-in-depth use cases can be broken down into user protection scenarios and network security scenarios.

Website protection

Defense-in-depth user protection involves a combination of security offerings (e.g., WAF, antivirus, antispam software, etc.) and training to block threats and protect critical data.

A vendor providing software to protect end-users from cyberattacks can bundle multiple security offerings in the same product. For example, packaging together antivirus, firewall, anti-spam and privacy controls.

As a result, the user's network is secured against malware, web application attacks (e.g., XSS, CSRF).

Network security

- An organization sets up a firewall, and in addition, encrypts data flowing through the network, and encrypts data at rest. Even if attackers get past the firewall and steal data, the data is encrypted.
- An organization sets up a firewall, runs an Intrusion Protection System with trained security operators, and deploys an antivirus program. This provides three layers of security – even if attackers get past the firewall, they can be detected and stopped by the IPS. And if they reach an end-user computer and try to install malware, it can be detected and removed by the antivirus.

SIEM

Security information and event management (SIEM) platforms collect log and event data from security systems, networks and computers, and turn it into actionable security insights. SIEM technology can help organizations detect threats that individual security systems cannot see, investigate past security incidents, perform incident response and prepare reports for regulation and compliance purposes.

Components and Capabilities in a SIEM Architecture

- 01 Data aggregation

  Collects and aggregates data from security systems and network devices

- 02 Threat intelligence feeds

  Combines internal data with third-party data on threats and vulnerabilities

- 03 Correlation and security monitoring

  Links events and related data into security incidents, threats or forensic findings

- 04 Analytics

  uses statistical models and machine learning to identify deeper relationships between data elements

- 05 Alerting

  Analyses events and sends alerts to notify security staff of immediate issues

- 06 Dashboards

  Creates visualizations to let staff review event data, identify patterns and anomalies

- 07 Compliance

  Gathers log data for standards like HIPAA, PCI/DSS, HITECH, SOX and GDPR and generates reports

- 08 Retention

Stores long-term historical data, useful for compliance and forensic investigations

- 09 Forensic analysis

  Enables exploration of log and event data to discover details of a security incident

- 10 Threat hunting

  Enables security staff to run queries on log and event data to proactively uncover threats

- 11 Incident response

  Helps security teams identify and respond to security incidents, bringing in all relevant data rapidly

- 12 SOC automation

  Advanced SIEMs can automatically respond to incidents by orchestrating security systems in an approach known as security orchestration and response (SOAR)

SIEM Logging Process

A SIEM server, at its root, is a log management platform. Log management involves collecting the data, managing it to enable analysis, and retaining historical data.

Data Collection

SIEMs collect logs and events from hundreds of organizational systems. Each device generates an event every time something happens, and collects the events into a flat log file or database. The SIEM can collect data in four ways:

1. Via an agent installed on the device (the most common method)
2. By directly connecting to the device using a network protocol or API call
3. By accessing log files directly from storage, typically in Syslog format
4. Via an event streaming protocol like SNMP, Netflow or IPFIX

The SIEM is tasked with collecting data from the devices, standardizing it and saving it in a format that enables analysis.

Data Management

SIEMs, especially at large organizations, can store mind-boggling amounts of data. The data needs to be:

- **Stored**—either on-premises, in the cloud or both
- **Optimized and indexed**—to enable efficient analysis and exploration
- **Tiered**—hot data necessary for live security monitoring should be on high-performance storage, whereas cold data, which you may one day want to investigate, should be relegated to high-volume inexpensive storage mediums

Log Retention

Industry standards like PCI DSS, HIPAA and SOX require that logs be retained for between 1 and 7 years. Large enterprises create a very high volume of logs every day from IT systems (see SIEM Sizing below). SIEMs need to be smart about which logs they retain for compliance and forensic requirements. SIEMs use the following strategies to reduce log volumes:

- **Syslog servers**—syslog is a standard which normalizes logs, retaining only essential information in a standardized format. Syslog lets you compress logs and retain large quantities of historical data.
- **Deletion schedules**—SIEMs automatically purge old logs that are no longer needed for compliance. By accessing log files directly from storage, typically in Syslog format.
- **Log filtering**—not all logs are needed for the compliance requirements faced by your organization, or for forensic purposes. Logs can be filtered by the source system, times, or by other rules defined by the SIEM administrator.
- **Summarization**—log data can be summarized to maintain only important data elements such as the count of events, unique IPs, etc.

Bypass an IDS (Intrusion Detection System)

Most intrusion detection systems work on a signature basis. It's quite possible for the attacker to create a custom packet payload that won't match any of the signatures in the predefined database of the IDS. This way, the attacker can bypass the IDS and possibly compromise the remote system without creating any noisy alerts.

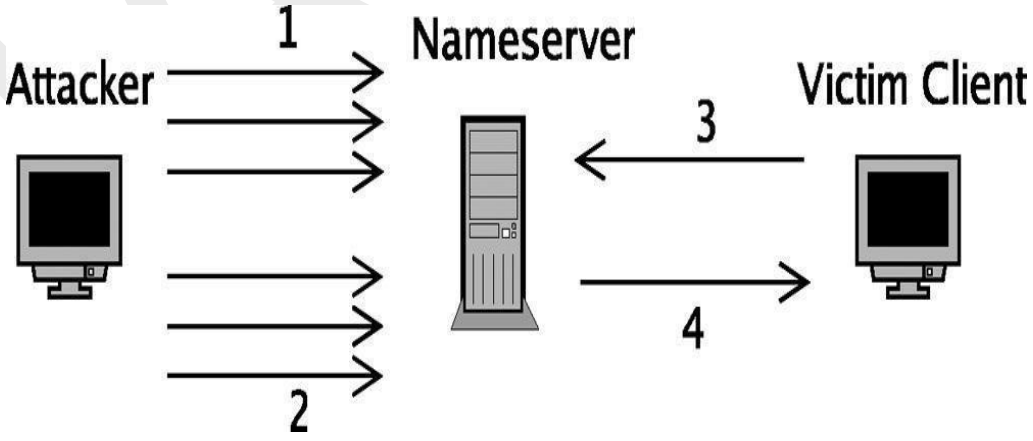
Following are some of the techniques used to evade and bypass an IDS.

Insertion attack:-

In an insertion attack, the attacker tries to confuse the IDS by sending invalid packets. The attacker crafts a malformed packet in such a way that the end system interprets the attack payload correctly but the IDS is unable to recognize the attack.

Denial of service :-

Many IDS systems use a centralized logging server to log all events and alerts. If the attackers know the IP address of this centralized logging server, they can launch a denial-of-service attack on that server so that the IDS won't be able to log any more events.



Obfuscating and encoding :-

Obfuscating means converting normal readable text or code into something that is hard to read and interpret. This is often used for security and privacy reasons. Encoding is a similar way of converting plain text into a special format and is mainly used for web transmissions.

Session splicing and fragmentation :-

Session splicing and fragmentation involve breaking, slicing, and splitting packets into multiple pieces such that no single packet causes the IDS to trigger an alert. Many IDS systems tend to ignore packet reconstruction before a packet is matched against the signature database.

Invalid packets :-

Sending invalid TCP packets is another way of evading an IDS. An attacker can manipulate one of the six TCP flags or the packet checksum in order to pass it through the IDS.

Polymorphic shellcodes :-

Most IDS systems have a standard default set of intrusion signatures. Attackers can modify the attack payload so that it doesn't match the default IDS signature and gets through it.

SNORT

SNORT is a network based intrusion detection system which is written in C programming language. It was developed in 1998 by Martin Roesch. Now it is developed by Cisco. It is a free open source software. It can also be used as a packet sniffer to monitor the system in real time. The network admin can use it to watch all the incoming packets and find the ones which are dangerous to the system. It is based on library packet capture tool. The rules are fairly easy to create and implement and it can be deployed in any kind on operating system and any kind of network environment. The main reason of popularity of this IDS over other is that it is a free to use software and also open source because of which any user can able to use it as the way he want.

Features:

- Real-time traffic monitor
- Packet logging
- Analysis of protocol
- Content matching
- OS fingerprinting
- Can be installed in any network environment.
- Creates logs
- Open Sourse
- Rules are easy to implement

Snort can be runned in 4 modes:

- sniffer mode: snort will read the network traffic and print them to the screen.
- packet logger mode: snort will record the network traffic on a file
- IDS mode: network traffic matching security rules will be recorded (mode used in our tutorial)
- IPS mode: also known as snort-inline (IPS = Intrusion prevention system)

Snort is a very powerful tool and is known to be one of the best IDS on the market even when comparedto commercial IDS.
A lot of people in the very active snort community are sharing their security rules which is very useful if you are not an security expert and wants to have up-to-date rules.
The SourceFire company is releasing very frequent new security rules that can be downloaded

either for free some days after their releases or immediatly but for money.

By chance, The bleedingsnort community create security rules for free directly after their releases.
Another tool is needed to display the logs generated by the Snort IDS and sent into the database. This tool is BASE for Basic Analysis and Security Engine. It is in fact a php script displaying alerts on a web interface.

Rule Header

alert – Rule action. Snort will generate an alert when the set condition is met.

any – Source IP. Snort will look at all sources.

any – Source port. Snort will look at all ports.

-> – Direction. From source to destination.

$HOME_NET – Destination IP. We are using the HOME_NET value from the snort.conf file.

any – Destination port. Snort will look at all ports on the protected network.

Rule Options

msg:”ICMP test” – Snort will include this message with the alert.

sid:1000001 – Snort rule ID. Remember all numbers < 1,000,000 are reserved, this is why we are starting with 1000001 (you may use any number, as long as it’s greater than 1,000,000).

rev:1 – Revision number. This option allows for easier rule maintenance.

classtype:icmp-event – Categorizes the rule as an “icmp-event”, one of the predefined

Nagios

Nagio is a free to use open source software tool for continuous monitoring. It helps you to monitor system, network, and infrastructure. It is used for continuous monitoring of systems, applications, service and business process in a DevOps culture.

Nagios runs plugins stored on the same server. It plugin's connects with a host or another server on your network or the Internet. Therefore, in the case of failure Nagios core can alert the technical staff about the issues. So that, your technical team performs the recovery process before outage in the business processes.

Why We Need Nagios?

Here, are Important reasons to use Nagios monitoring tool are:

- Detects all types of network or server issues
- Helps you to find the root cause of the problem which allows you to get the permanent solution to the problem
- Active monitoring of your entire infrastructure and business processes
- Allows you to monitors and troubleshoot server performance issues
- Helps you to plan for infrastructure upgrades before outdated systems create failures
- You can maintain the security and availability of the service
- Automatically fix problems in a panic situation

History of Nagios

1996-Ethan Galstad uses the ideas and architecture of his earlier work to begin building a new application which runs under Linux OS

1999-The plugins that were which were originally distributed as a part of the NetSaint distribution are soon as a separate Nagios Plugins project

2002- Ethan renames the project to "Nagios" because of trademark issues with the name "NetSaint."

2005- Nagios becomes SourceForge.net Project of the Month in June

2009-Nagios Enterprises releases its first commercial version, Nagios XI

2012-Nagios again renamed as Nagios Core

2016-Nagios core surpasses 7,500,000 downloads directly from SourceForge.net website

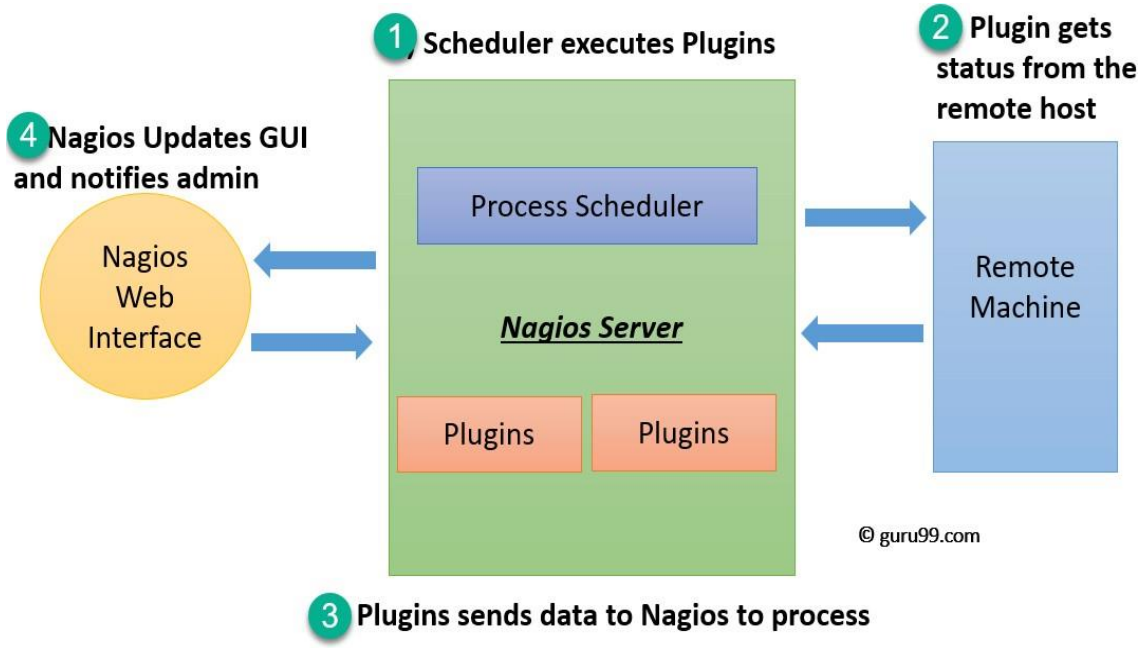
Features of Nagios

Following are the important features of Nagios:

- Relatively scalable, Manageable, and Secure
- Good log and database system
- Informative and attractive web interfaces
- Automatically send alerts if condition changes
- If the services are running fine, then there is no need to do check that host is an alive
- Helps you to detect network errors or server crashes
- You can troubleshoot the performance issues of the server.
- The issues, if any, can be fixed automatically as they are identified during the monitoring process
- You can monitor the entire business process and IT infrastructure with a single pass
- The product's architecture is easy writing new plugins in the language of your choice
- Nagios allows you to read its configuration from an entire directory which helps you to decide how to define individual files
- Utilizes topology to determine dependencies
- Monitor network services like HTTP, SMTP, HTTP, SNMP, FTP, SSH, POP, etc.
- Helps you to define network host hierarchy using parent hosts
- Ability to define event handlers which runs during service or host events for proactive problem resolution
- Support for implementing redundant monitoring hosts

Nagios Architecture

Nagios is a client-server architecture. Usually, on a network, a Nagios server is running on a host, and plugins are running on all the remote hosts which should be monitored.

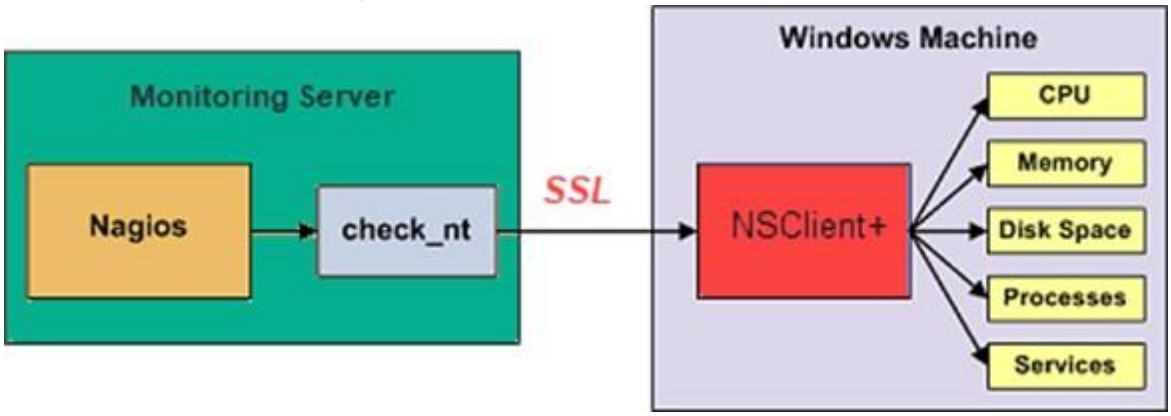


1. The scheduler is a component of server part of Nagios. It sends a signal to execute the plugins at the remote host.
2. The plugin gets the status from the remote host
3. The plugin sends the data to the process scheduler
4. The process scheduler updates the GUI and notifications are sent to admins

Plugins:

Nagios plugins provide low-level intelligence on how to monitor anything and everything with Nagios Core. Plugins operate acts as a standalone application, but they are designed to be executed by Nagios Core. It connects to Apache that is controlled by CGI to display the result. Moreover, a database connected to Nagios to keep a log file.

How do plugins work?



Consider the above example-

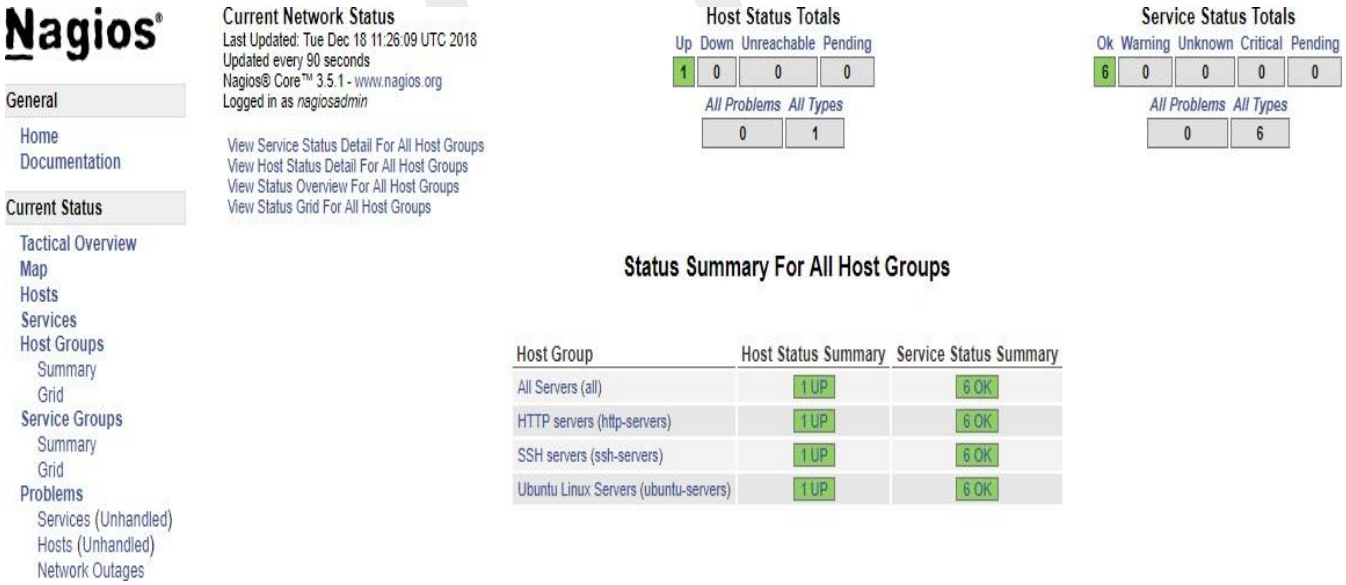
- Check_nt is a plugin to monitor a windows machine which is mostly available in themonitoring server
- NSClinet++ should be installed in every Windows machine that you wants to monitor
- There is an SSL connection between the server and the host which continuouslyexchange information with each other

Likewise, NRPE(Nagios Remote plug-in Executor) and NSCA plugins are used to monitor Linuxand Mac OS X respectively.

GUI:

An interface of Nagios is used to display in web pages generated by CGI. It can be buttons togreen or red, sound, graph, etc.

When the soft alert is raised many times, a hard alert is raised, then the Nagios server sends anotification to the administrator.



Nagios GUI

Introduction to Information security

Information Security is not only about securing information from unauthorized access. Information Security is basically the practice of preventing unauthorized access, use, disclosure, disruption, modification, inspection, recording or destruction of information. Information can be physical or electronic one. Information can be anything like Your details or we can say your profile on social media, your data in mobile phone, your biometrics etc. Thus, Information Security spans so many research areas like Cryptography, Mobile Computing, Cyber Forensics, online social media etc.

Why is information security important?

Whichever type of information security management provider you choose, the quality of the security measures is essential. You need to be confident that you're protected from unauthorized access and security breaches. The device and network security services should cover the following areas:

- Reducing the risk of data breaches and attacks in IT systems.
- Applying security controls to prevent unauthorized access to sensitive information.
- Preventing disruption of services, e.g., denial-of-service attacks.
- Protecting IT systems and networks from exploitation by outsiders.
- Keeping downtime to a minimum so productivity stays high.
- Ensuring business continuity through data protection of information assets.
- Providing peace of mind by keeping confidential information safe from securitythreats.
- information security is various measures to protect information from unauthorized persons. In the pre-digital era, people locked important documents in safes, hired security guards, and encrypted their messages on paper to protectdata.
- Today, digital information is more often protected. Still, the measures

are essentially the same: information security specialists create protected spaces (virtual "safes"), install security software like antivirus ("hire security guards") and use cryptographic methods to encrypt digital information.

7 Different Types of Firewalls

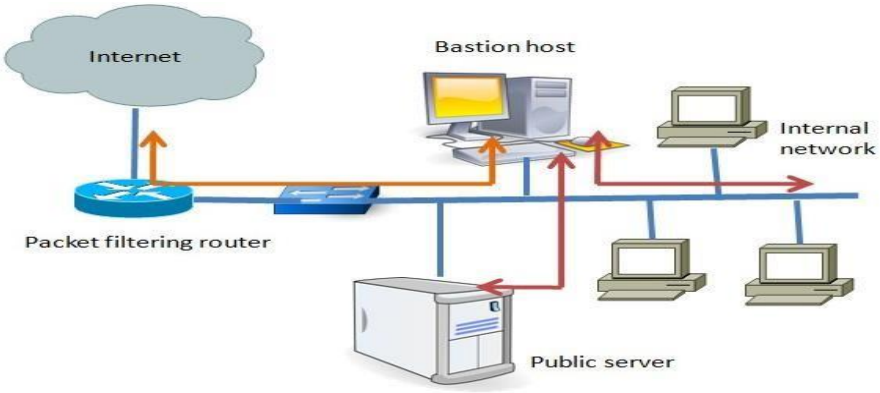
There are several types of firewalls that work on different layers of the OSI model. Depending on the kind of service and security you need for your network, you need to choose the right type of firewall. The following are the list of seven different types firewalls that are widely used for networksecurity.

- Screened host firewalls
- Screened subnet firewalls
- Packet filter firewalls
- Stateful inspection firewalls
- Hybrid firewalls
- Proxy server firewalls
- Application level (gateway) firewalls

1. Screened host firewalls:

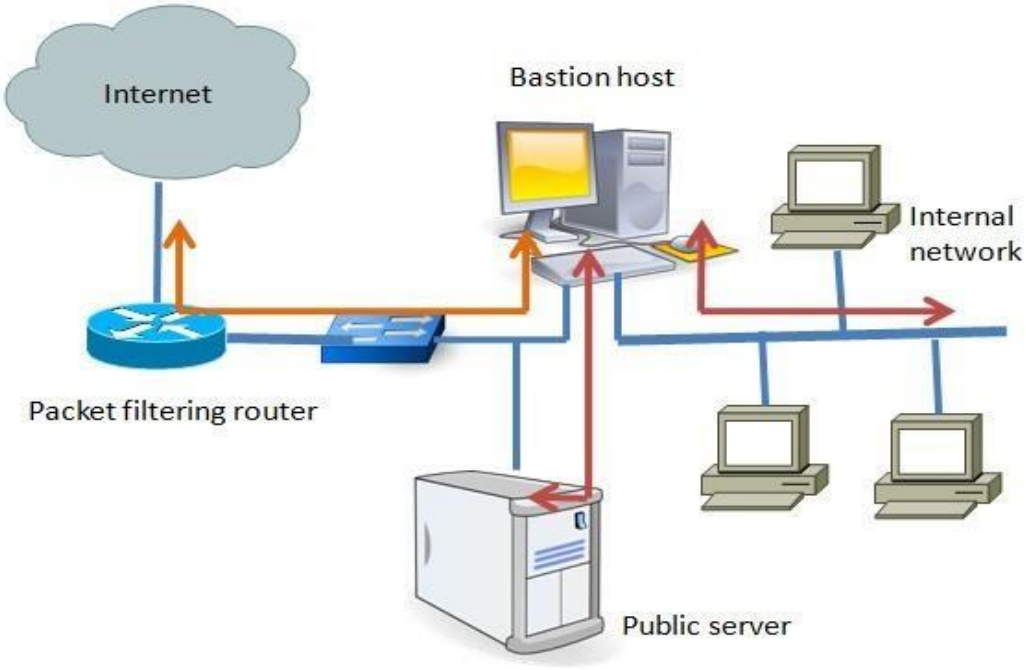
There are two types of screened host-one is single homed bastion host and the other one is dual homed bastion host**.** In case of single homed bastion host the firewall system consists of a packet filtering router and a bastion host. A bastion host is basically a single computer with high security configuration, which has the following characteristics:

- Traffic from the Internet can only reach the bastion host; they cannotreach the internal network.
- Traffic having the IP address of the bastion host can only go to theInternet. No traffic from the internal network can go to the Internet.



Screened host firewall (single-homed bastion host)

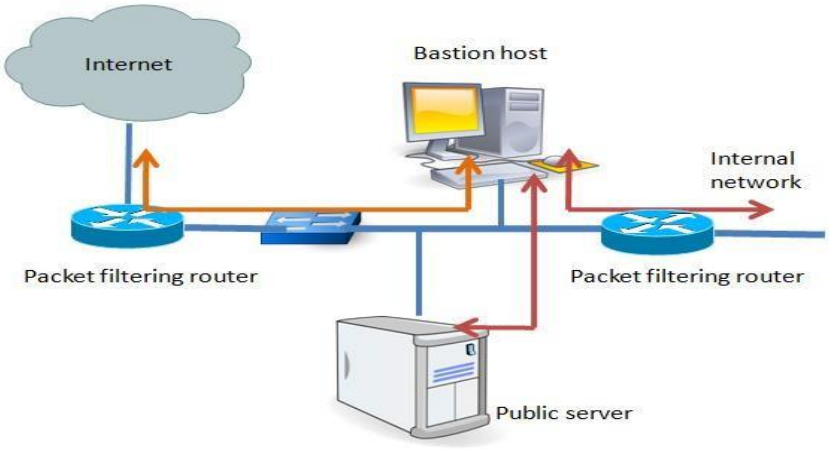
This type of configuration can have a web server placed in between the routerand the bastion host in order to allow the public to access the server from theInternet. The main problem with the single homed bastion host is that if the packet filter route gets compromised then the entire network will be compromised. To eliminate this drawback, we can use the dual homed bastionhost firewall system, where a bastion host has two network cards- one is usedfor internal connection and the second one is used for connection with the router. In this case, even if, the router got compromised, the internal networkwill remain unaffected since it is in the separate network zone.



Screened host firewall (Dual-homed bastion host)

2. Screened subnet firewalls

This is one of the most secured firewall configurations. In this configuration, two packet filtering routers are used and the bastion host is positioned in between the two routers. In a typical case, both the Internet and the internal users have access to the screened subnet, but the traffic flow between the two subnets (one is from bastion host to the internal network and the other is the sub-network between the two routers) is blocked.



Screened subnet firewall

3. Packet filtering firewalls

Packet filtering is a firewall technique used to control network access by monitoring outgoing and incoming packets and allowing them to pass or halt based on the source and destination Internet Protocol (IP) addresses, protocols and ports.

Network layer firewalls define packet filtering rule sets, which provide highly efficient security mechanisms. Packet filtering is also known as static filtering. During network communication, a node transmits a packet that is filtered and matched with predefined rules and policies. Once matched, a packet is either accepted or denied.

Packet filtering checks source and destination IP addresses. If both IP addresses match, the packet is considered secure and verified. Because the sender may use different applications and programs, packet filtering also checks source and destination protocols, such as User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). Packet filters also verify source and destination port addresses.

Some packet filters are not intelligent and unable to memorize used packets.

However, other packet filters can memorize previously used packet items, such as source and destination IP addresses.

Packet filtering is usually an effective defense against attacks from computers outside a local area network (LAN). As most routing devices have integrated filtering capabilities, packet filtering is considered a standard and cost-effective means of security.

This type of firewall is the most common and easy to deploy in a small-sized network. A router functions as a firewall by examining every packet passing through the network. Based on access control list, the router either forward or drop packets. Normally, the IP address of the source and destination, port number and type of traffic are taken into account when the router processes each data packet. Since a router cannot check packet in the application layer, this type of firewall cannot defend attacks that use application layers vulnerabilities. They also fail to fight against spoofing attacks. You can use this configuration if you need higher network speed and do need limited login and authentication capacity.

4. Stateful inspection

Stateful inspection firewall works at the network layer in the OSI model. It monitors both the header and contents of the traffic. The main difference between the packet filtering and the stateful inspection is that it the later one analyses not only the packet headers but also inspects the state of the packets along with providing proxy services. Stateful inspection firewalls maintain a state table and a set of instructions to inspect each packet and store the information based on the type of traffic. It also monitors each TCP connection and remembers which ports are being used by that connection. If there is any port not required by the connection, then that port get closed.

5. Hybrid firewalls

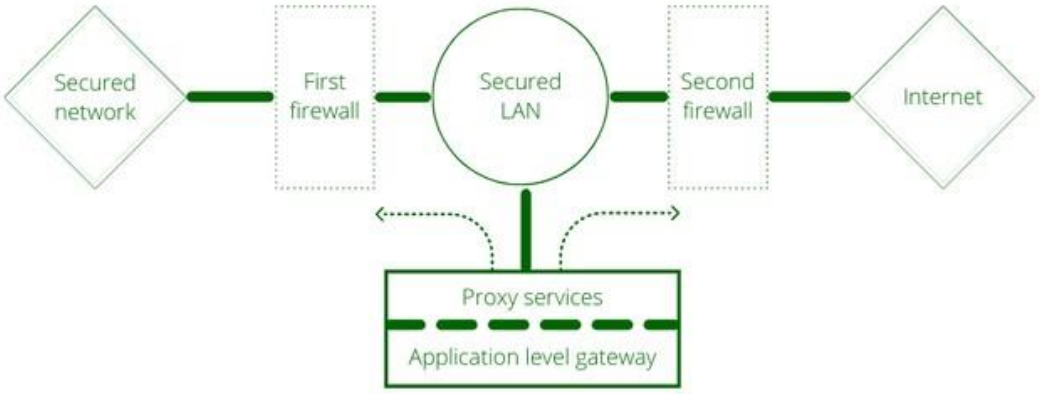
They function almost the same way the stateful inspection type firewalls work, which means they can work both in network and in application level. Normally, in a hybrid system some hosts reside inside the firewall while the others reside outside of the firewall. To communicate with the machine outside the central network IPsec tunnels are used. An example where this type of configuration is suitable is a major site connected with its branch sites via VPN. One

distinct feature of this configuration is the firewall administration at the major site distribute the security policy to its branch site so as a uniform security is maintained throughout the organization.

6. Proxy server firewalls

Proxy allows users to run specific service (FTP, TELNET, HTTP etc.) or type of connection by enforcing authentication, filtering and logging. For specific service there will be a specific proxy. For example, if you want to allow only HTTP connection to the Internet for your internal network users, then you must allow only HTTP proxy, nothing else. Users who need to go to Internet create a virtual circuit with the proxy server and the proxy server sends the request to connect to a specific site on behalf of that particular user. Proxy server changes the IP of the request so as the Internet or the outside world can see only the IP of the proxy server. Thus, proxy server hides the internal network behind it**.** When a proxy receives the data from the Internet it sends the data back to its intended internal user via the virtual circuit. The main advantage of using proxy is that it is fully aware of the type of data it handles and can give protection to it. One disadvantage of proxy is that if there is an update of protocol that is used by the Internet, then the proxy software also needs to be updated to allow a specific service related to that protocol.

7. Application level (gateway) firewalls



Application-level gateway is also called a bastion host. It operates at the

application level. Multiple application gateways can run on the same

host but each gateway is a separate server with its own processes.

These firewalls, also known as application proxies, provide the most secure type of data connection because they can examine every layer of the communication, including the application data.

Example: Consider FTP service. The FTP commands like getting the file, putting the file, listing files, and positioning the process at a particular point in a directory tree. Some system admin blocks put command but permits get command, list only certain files, or prohibit changing out of a particular directory. The proxy server would simulate both sides of this protocol exchange. For example, the proxy might accept get commands and reject put commands.

It works as follows:

Step-1: User contacts the application gateway using a TCP/IP application such as HTTP.

Step-2: The application gateway asks about the remote host with which the userwants to establish a connection. It also asks for the user id and password that is required to access the services of the application gateway.

Step-3: After verifying the authenticity of the user, the application gateway accesses the remote host on behalf of the user to deliver the packets.

Difference between Packet filter & Application-level

Packet filter	**Application-level**
Simplest	Even more complex
Screens based on connection rules	Screens based on behaviour or proxies
Auditing is difficult	Activity can audit
Low impact on network performance	High impact on network performance

Network topology cannot hide	Network topology can hide from the attacker
Transparent to user	Not transparent to the user
See only addresses and serviceprotocol type	Sees full data portion of a packet

What is a next-generation firewall?

A traditional firewall provides stateful inspection of network traffic. It allows or blocks traffic based on state, port, and protocol, and filters traffic based on administrator-defined rules.

A next-generation firewall (NGFW) does this, and so much more. In addition to access control, NGFWs can block modern threats such as advanced malware and application-layer attacks. According to Gartner's definition, a next-generation firewall must include:

- Standard firewall capabilities like stateful inspection
- Integrated intrusion prevention
- Application awareness and control to see and block risky apps
- Threat intelligence sources
- Upgrade paths to include future information feeds
- Techniques to address evolving security threats

Wireshark

Wireshark is a free and open-source packet analyser. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Originally named **Ethereal**, the project was renamed Wireshark in May 2006 due to trademark issues.

Wireshark is absolutely safe to use. Government agencies, corporations, non-profits, and educational institutions use Wireshark for troubleshooting and teaching purposes. There isn't a better way to learn networking than tolook at the traffic under the Wireshark microscope.

There are questions about the legality of Wireshark since it is a powerful packet sniffer. The Light side of the Force says that you should only use

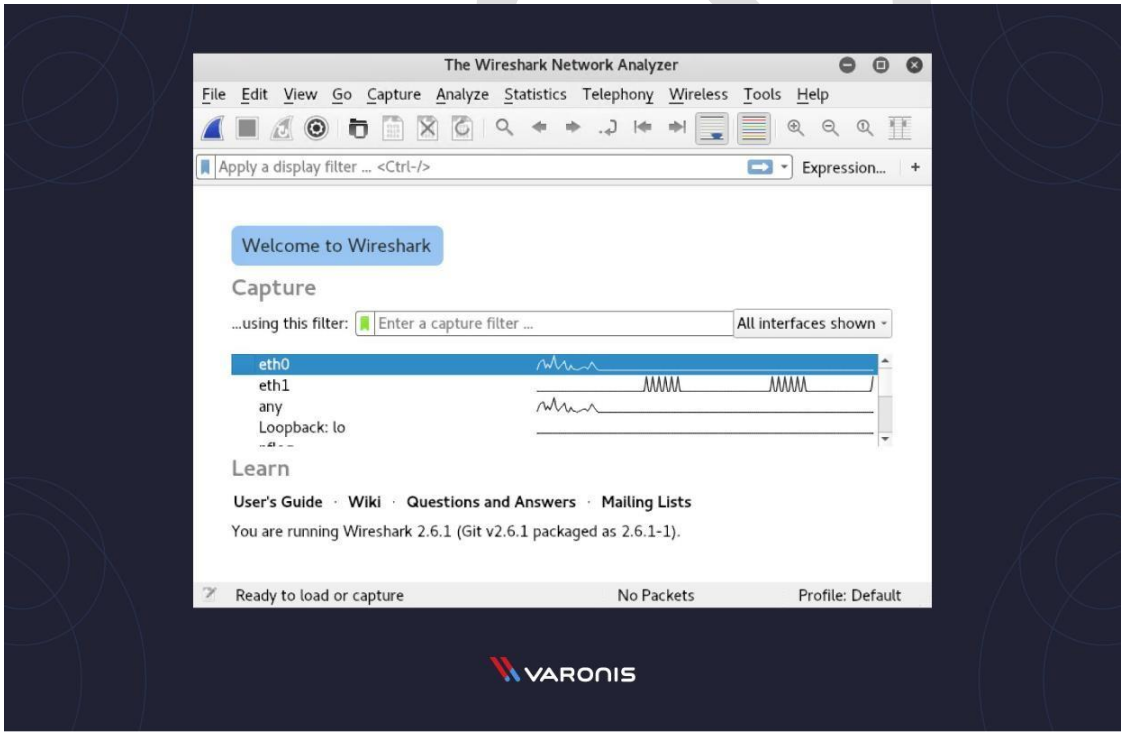
Wireshark on networks where you have permission to inspect network packets. Using Wireshark to look at packets without permission is a path to the Dark Side.

Wireshark is very similar to tcpdump, but has a graphical front-end and integrated sorting and filtering options.

Wireshark lets the user put network interface controllers into promiscuous mode (if supported by the network interface controller), so they can see allthe traffic visible on that interface including unicast traffic not sent to that network interface controller's MAC address. However, when capturing with a packet analyser in promiscuous mode on a port on a network switch, not all traffic through the switch is necessarily sent to the port where the capture is done, so capturing in promiscuous mode is not necessarily sufficient to see all network traffic.

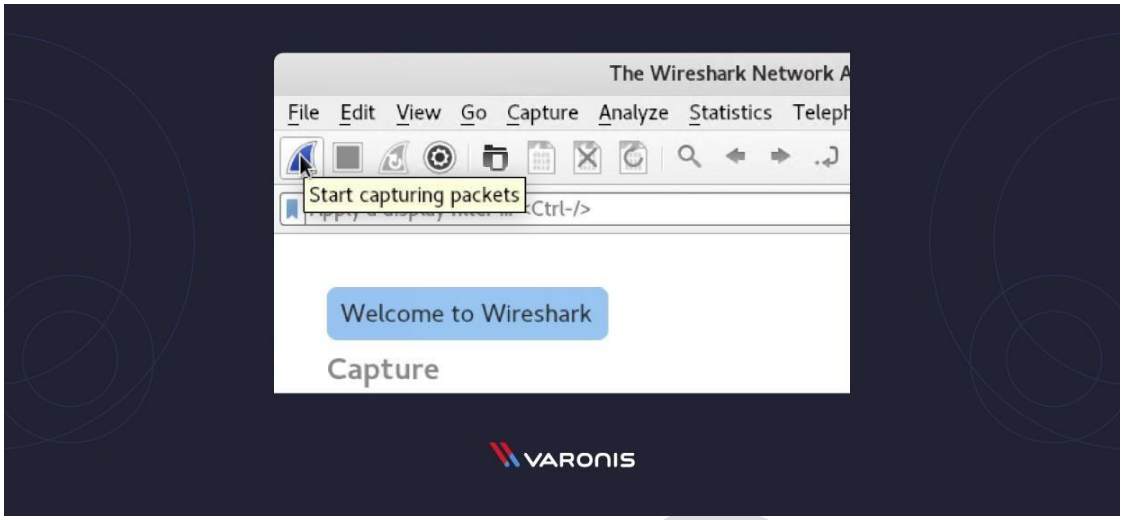
Capturing Data Packets on Wireshark

When you open Wireshark, you see a screen that shows you a list of all ofthe network connections you can monitor. You also have a capture filter field, so you only capture the network traffic you want to see.

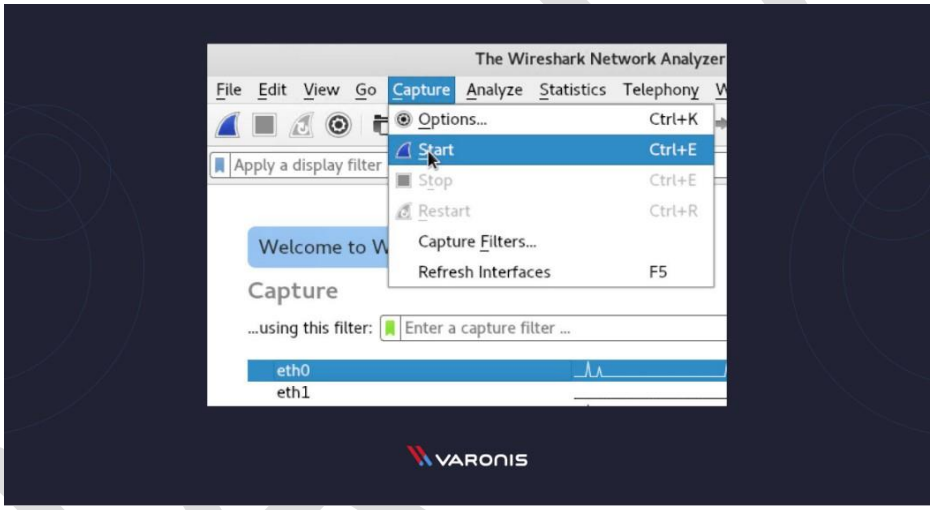


You can select one or more of the network interfaces using “shift left- click.” Once you have the network interface selected, you can start the capture, and there are several ways to do that.

Click the first button on the toolbar, titled “Start Capturing Packets.”

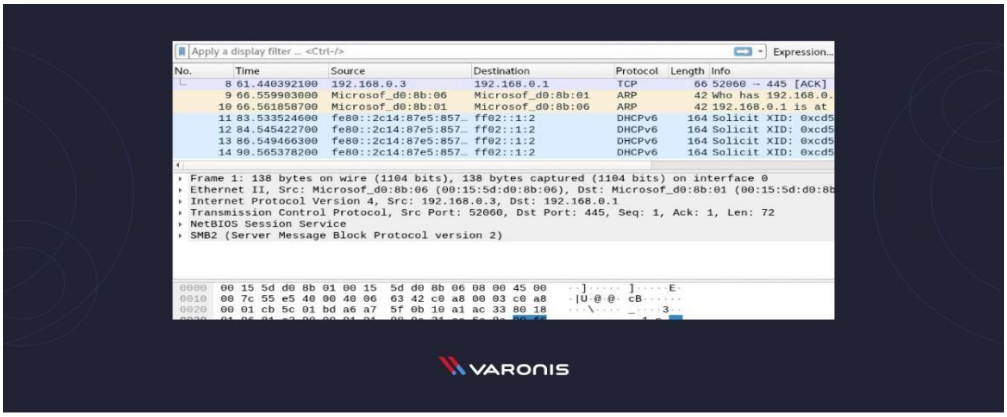


You can select the menu item Capture -> Start.



Or you could use the keystroke Control – E.

During the capture, Wireshark will show you the packets that it captures inreal-time.



Once you have captured all the packets you need, you use the same buttonsor menu options to stop the capture.

Best practice says that you should stop Wireshark packet capture before you do analysis.

Wireshark Filters

One of the best features of Wireshark is the Wireshark Capture Filters and Wireshark Display Filters. Filters allow you to view the capture the way you need to see it so you can troubleshoot the issues at hand. Here are several filters to get you started.

Wireshark Capture Filters

Capture filters limit the captured packets by the filter. Meaning if the packets don't match the filter, Wireshark won't save them. Here are someexamples of capture filters:
host IP-*address*: this filter limits the capture to traffic to and from the IP address
net 192.168.0.0/24: this filter captures all traffic on the subnet.

dst host IP-*address*: capture packets sent to the specified host.port
53: capture traffic on port 53 only.

port not 53 and not arp: capture all traffic except DNS and ARP traffic

Wireshark Display Filters

Wireshark Display Filters change the view of the capture during analysis. After you have stopped the packet capture, you use display filters to narrowdown the packets in the Packet List so you can troubleshoot your issue.
The most useful (in my experience) display filter is:

ip.src==*IP-address* and ip.dst==*IP-address*

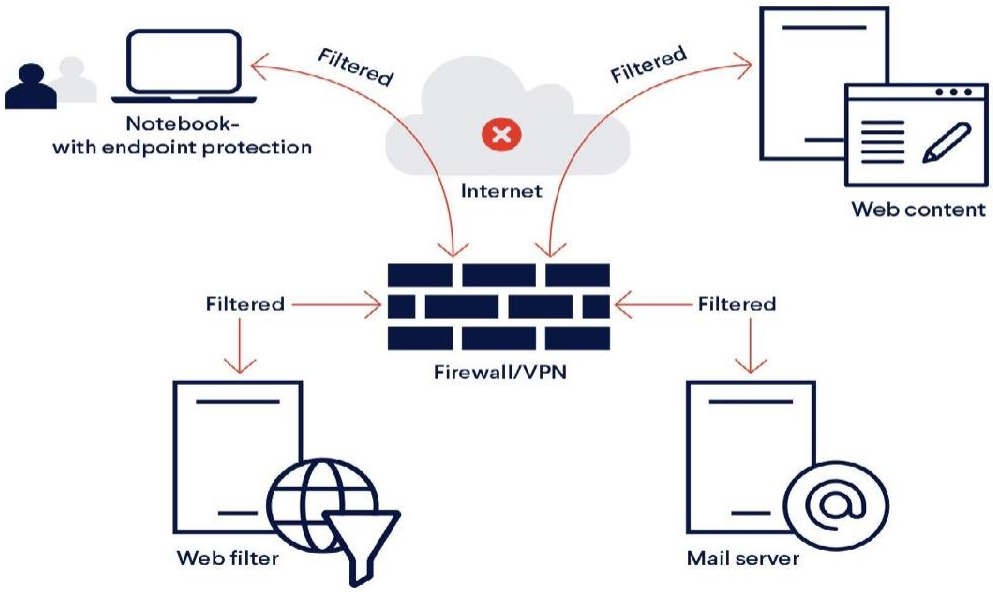
This filter shows you packets from one computer (ip.src) to another (ip.dst). You can also use ip.addr to show you packets to and from that IP. Here are some others:

tcp.port eq 25: This filter will show you all traffic on port 25, which isusually SMTP traffic.

icmp: This filter will show you only ICMP traffic in the capture, mostlikely they are pings.

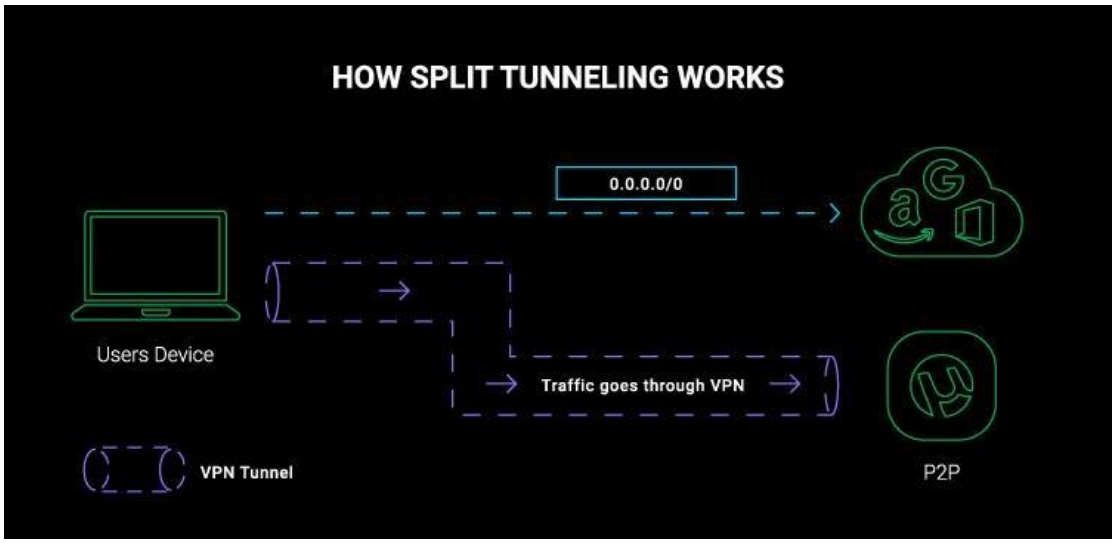
ip.addr != *IP_address*: This filter shows you all traffic except the traffic toor from the specified computer.
Analysts even build filters to detect specific attacks, like this filter to detectthe Sasser worm:
ls_ads.opnum==0x09

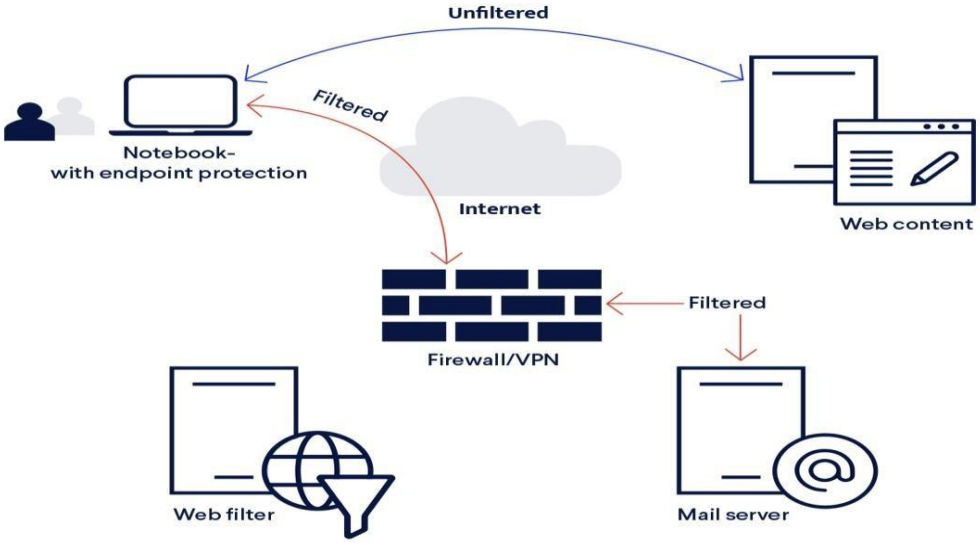
Full-tunnel



In a full-tunnel VPN scenario, whenever the user connects to the enterprise network, all network connections go through the enterprise network. Whenever the user starts a new YouTube video or Netflix movie, all network packets traverse through the enterprise network. Supporting this scenario for all employees might involve upgrading costly business Internet connection lines, network equipment, VPN servers, etc.

Split-tunnel VPN:





Split tunnelling works by giving you **two connections at the same time**: the secure VPN connection and an open connection to the internet. So, you can protect your sensitive data without slowing down your other internet activities.

Split tunnelling is a VPN feature that **divides your internet traffic** and sends some of it through an encrypted virtual private network (VPN) tunnel, but routes the rest through a separate tunnel on the open network. Typically, split tunnelling will let you choose which apps to secure and which can connect normally.

This is a useful feature when you need to keep some of your traffic private,while still **maintaining access to local network** devices.

So, you can access foreign networks and local networks at the same time. It's also great if you want to save some bandwidth.

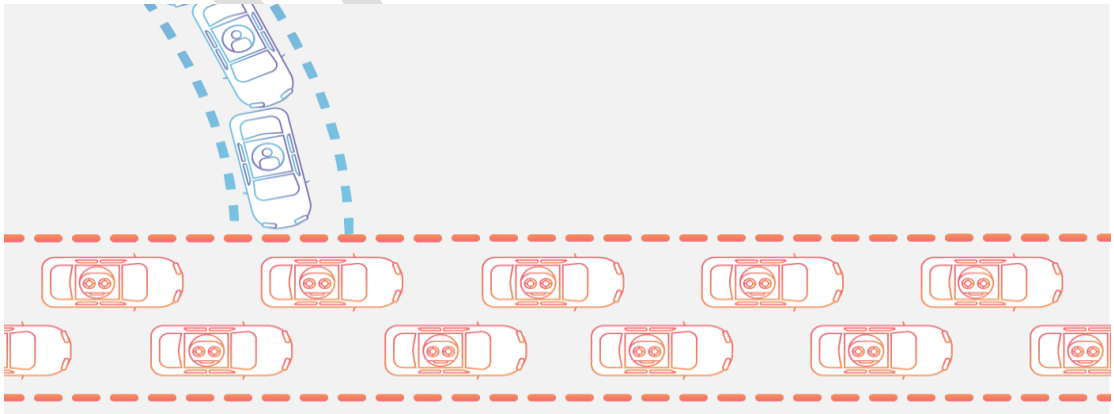
In a split-tunnel VPN scenario, only packets where the destination is in the company are routed to the company network. IT teams must set up new VPN connections for many users recently, and many IT teams choose a split tunnel. In many cases, companies had to switch from full-tunnel VPN to split tunnel due to infrastructure that is incapable of working under the extensive full-tunnel VPN load. As there is no one-size-fits-all in risk management, each company should calculate the cost difference between the full- and split-tunnel VPN scenarios, and measure this against the increased risks of malware infection, phishing attacks, etc.

Attacks-distributed

A distributed denial-of-service (DDoS) attack is a malicious attempt to disrupt the normal traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic.

DDoS attacks achieve effectiveness by utilizing multiple compromised computer systems as sources of attack traffic. Exploited machines can include computers and other networked resources such as IoT devices.

From a high level, a DDoS attack is like an unexpected traffic jam clogging up the highway, preventing regular traffic from arriving at its destination.



Intruder types

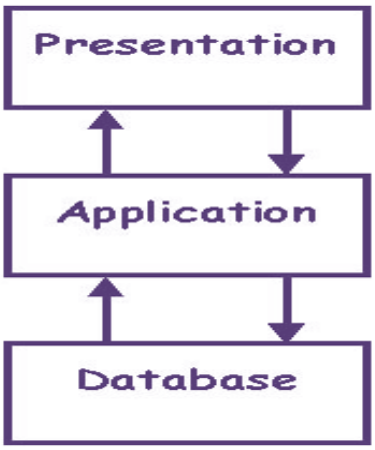
Intruders are the attackers who attempt to breach the security of a network. They attack the network in order to get unauthorized access. Intruders areof three types, namely, *masquerader*, *misfeasor* and *clandestine user*.

Masquerader: pretend to be someone one is not An individual who is not authorized to use the computer and who penetrates a system's access controls to exploit a legitimate user's account **Misfeasor**: authentic user doing unauthorized actions A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses his or her privileges

Clandestine user: done secretively, especially because illicitAn individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection

Tired Architecture

What is N-Tier? An N-Tier Application program is one that is distributed among three or more separate computers in a distributed network. The most common form of n-tier is the 3-tier Application, and it is classified into three categories. • User interface programming in the user's computer • Business logic in a more centralized computer, and • Required data in a computer that manages a database. This architecture model provides Software Developers to create Reusable application/systems with maximum flexibility. In N-tier, "N" refers to a number of tiers or layers arebeing used like – 2-tier, 3-tier or 4-tier, etc. It is also called "Multitier Architecture". The n-tier architecture is an industry proven software architecture model. It is suitable to support enterprise level client- server applications by providing solutions to scalability, security, fault tolerance,reusability, and maintainability. It helps developers to create flexible and reusable applications. N-Tier Architecture A diagrammatic representationof an n-tier system depicts here – presentation, application, and database layers.



These three layers can be further subdivided into different sub-layersdepending on the requirements.

Some of the popular sites who have applied this architecture are

- MakeMyTrip.com
- Sales Force enterprise application
- Indian Railways – IRCTC
- Amazon.com, etc.