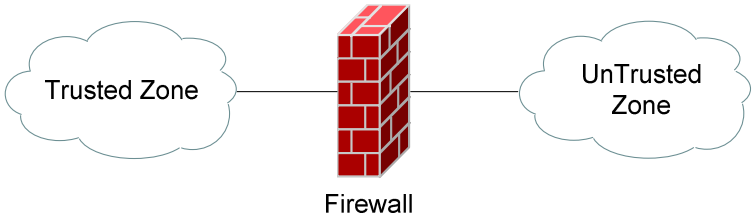
- Introduction to Firewalls -

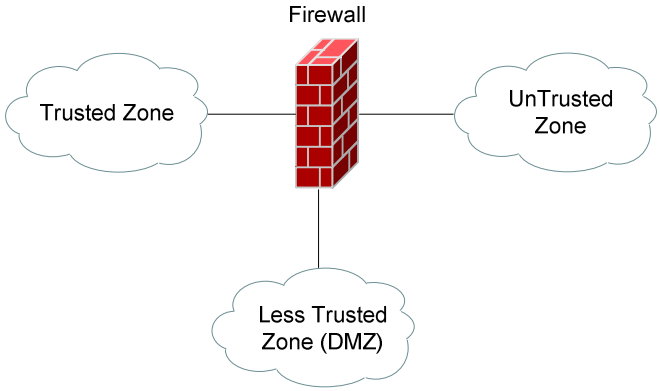
Firewall Basics

Traditionally, a firewall is defined as any device (or software) used to filter or control the flow of traffic. Firewalls are typically implemented on the network perimeter, and function by defining **trusted** and **untrusted zones**:



Most firewalls will **permit** traffic from the *trusted* zone to the *untrusted* zone, **without** any explicit configuration. However, traffic from the *untrusted* zone to the *trusted* zone must be **explicitly permitted.** Thus, any traffic that is not explicitly permitted from the untrusted to trusted zone will be **implicitly denied** (by default on *most* firewall systems)**.**

A firewall is not limited to only two zones, but can contain multiple 'less trusted' zones, often referred to as **Demilitarized Zones (DMZ's).**



To control the *trust* value of each zone, each firewall interface is assigned a *security level*, which is often represented as a numerical value or even color. For example, in the above diagram, the Trusted Zone could be assigned a security value of 100, the Less Trusted Zone a value of 75, and the Untrusted Zone a value of 0.

As stated previously, traffic from a *higher* security to *lower* security zone is (generally) allowed by default, while traffic from a *lower* security to *higher* security zone requires explicit permission.

* * *

Firewall Services

Firewalls perform the following services:

- **Packet Filtering**
- **Stateful Packet Inspection**
- **Proxying**
- **Network Address Translation (NAT)**

Each will be covered in some detail in this guide.

Packet Filtering

Packet Filtering is one of the core services provided by firewalls. Packets can be filtered (*permitted* or *denied*) based on a wide range of criteria:

- Source address
- Destination address
- Protocol Type (IP, TCP, UDP, ICMP, ESP, etc.)
- Source Port
- Destination Port

Packet filtering is implemented as a **rule-list**:

Number	Action	Protocol	Source Add.	Source Port	Destination Add.	Destination Port
1.	Deny	TCP	Any	Any	172.16.1.5	666
2.	Permit	IP	Any	Any	172.16.1.5	Any
3.	Permit	TCP	Any	Any	172.16.1.1	443
4.	Permit	TCP	Any	Any	172.16.1.1	80
5.	Permit	TCP	Any	Any	172.16.1.10	25
6.	Deny	TCP	66.1.1.5	Any	172.16.1.10	110
7.	Permit	TCP	Any	Any	172.16.1.10	110

The *order* of the rule-list is a critical consideration. The rule-list is *always* parsed from **top-to-bottom**. Thus, more specific rules should always be placed near the *top* of the rule-list, otherwise they may be negated by a previous, more encompassing rule.

Also, an implicit 'deny any' rule usually exists at the bottom of a rule-list, which often can't be removed. Thus, rule-lists that contain **only deny statements** will **prevent all traffic**.

* * *

Stateful Packet Inspection

Stateful packet inspection provides services beyond simple packet-filtering, by additionally *tracking* TCP or UDP sessions between devices.

For example, stateful inspection can track connections that originate from the *trusted* network. This session information is kept in a **state session table**, which allows temporary holes to be opened in the firewall for the *return traffic*, which might otherwise be denied.

Connections from the *untrusted* network to the *trusted* network are also monitored, to prevent Denial of Service (DoS) attacks. If a high number of **half-open sessions** are detected, the firewall can be configured to drop the session (and even block the source), or send an alert message indicating an attack is occurring.

A **half-open TCP** session indicates that the three-way handshake has not yet completed. A **half-open UDP** session indicates that no return UDP traffic has been detected. A large number of half-opened sessions will chew up resources, while preventing legitimate connections from being established.

Proxy Services

A proxy server, by definition, is used to make a request *on behalf* of another device. It essentially serves as a middle-man for communication between devices.

This provides an element of security, by hiding the actual requesting source. All traffic will seem to be originated from the proxy itself.

Traditionally, proxy servers were used to **cache** a local copy of requested external data. This improved performance in limited-bandwidth environments, allowing clients to request data from the *proxy*, instead of the actual *external source*.

Other services that proxy servers can provide:

- Logging
- Content Filtering
- Authentication

NAT (Network Address Translation)

The rapid growth of the Internet resulted in a shortage of IPv4 addresses. In response, the powers that be designated a specific subset of the IPv4 address space to be *private,* to temporarily alleviate this problem.

A **public address** can be routed on the Internet. Thus, devices that should be Internet accessible (such web or email servers) must be configured with public addresses.

A **private address** is only intended for use within an organization, and can never be routed on the internet. Three private addressing ranges were allocated, one for each IPv4 class:

- Class A - **10.x.x.x**
- Class B - **172.16-31.x.x**
- Class C - **192.168.x.x**

NAT (Network Address Translation) is used to translate between private addresses and public addresses. NAT allows devices configured with a private address to be *stamped* with a public address, thus allowing those devices to communicate across the Internet.

NAT is *not* restricted to just public-to-private address translations, though this is the most common application of NAT. NAT can perform a public-to-public address translation, or a private-to-private address translation as well.

NAT provides an additional benefit – hiding the specific addresses and addressing structure of the internal network.

(Reference: http://www.cisco.com/en/US/tech/tk648/tk361/technologies_white_paper09186a0080194af8.shtml)

* * *

Types of NAT

NAT can be implemented using one of three methods:

Static NAT – performs a static one-to-one translation between two addresses, or between a *port* on one address to a port on another address. Static NAT is most often used to assign a public address to a device behind a NAT-enabled firewall/router.

Dynamic NAT – utilizes a **pool** of global addresses to dynamically translate the outbound traffic of clients behind a NAT-enabled device.

NAT Overload or **Port Address Translation (PAT) –** translates the outbound traffic of clients to unique port numbers off of a *single* global address. PAT is necessary when the number of internal clients exceeds the available global addresses.

NAT Terminology

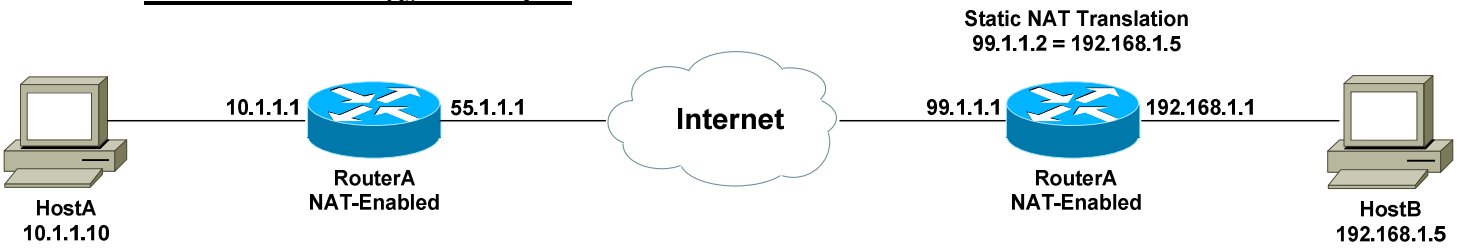
Specific terms are used to identify the various NAT addresses:

- **Inside Local –** the specific IP address assigned to an *inside* host behind a NAT-enabled device (usually a *private* address).
- **Inside Global –** the address that identifies an *inside* host to the *outside* world (usually a *public* address). Essentially, this is the dynamically or statically-assigned public address assigned to a private host.
- **Outside Global** – the address assigned to an *outside* host (usually a *public* address).
- **Outside Local** – the address that identifies an *outside* host to the *inside* network. Often, this is the **same** address as the Outside Global. However, it is occasionally necessary to translate an outside (usually *public*) address to an inside (usually *private*) address.

For simplicity sake, it is generally acceptable to associate **global** addresses with **public** addresses, and **local** addresses with **private** addresses. However, remember that public-to-public and private-to-private translation is still possible. **Inside** hosts are within the local network, while **outside** hosts are external to the local network.

* * *

NAT Terminology Example



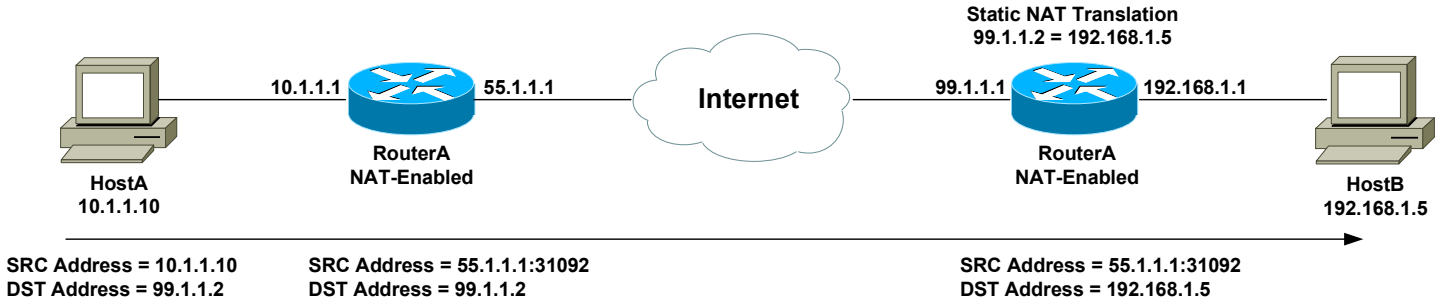
Consider the above example. For a connection *from* HostA *to* HostB, the NAT addresses are identified as follows:

- **Inside Local Address** - 10.1.1.10
- **Inside Global Address** - 55.1.1.1
- **Outside Global Address** – 99.1.1.2
- **Outside Local Address –** 99.1.1.2

HostA's configured address is *10.1.1.10*, and is identified as its *Inside Local* address. When HostA communicates with the Internet, it is stamped with RouterA's public address, using PAT. Thus, HostA's *Inside Global* address will become *55.1.1.1*.

When HostA communicates with HostB, it will access HostB's *Outside Global* address of *99.1.1.2*. In this instance, the *Outside Local* address is also *99.1.1.2*. HostA is never aware of HostB's configured address.

It is possible to map an address from the local network (such as 10.1.1.5) to the global address of the remote device (in this case, 99.1.1.2). This may be required if a legacy device exists that will only communicate with the local subnet. In this instance, the *Outside Local* address would be *10.1.1.5*.



The above example demonstrates how the source (SRC) and destination (DST) IP addresses within the Network-Layer header are translated by NAT.

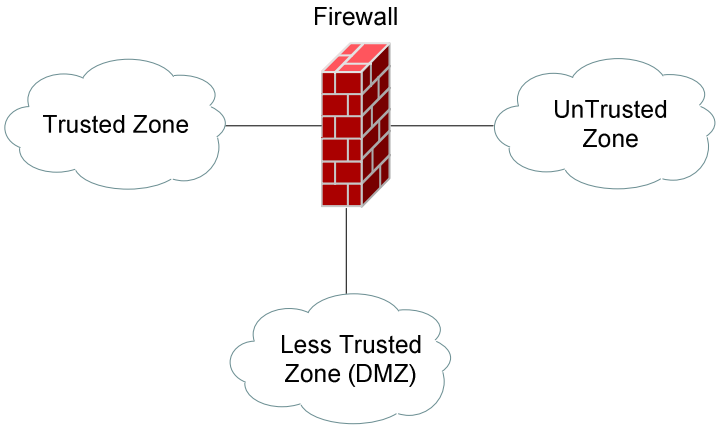
(Reference: *http://www.cisco.com/warp/public/556/8.html*)

* * *

Implementing a DMZ

As briefly described earlier, a DMZ is essentially a **less trusted zone** that sits between the **trusted zone** (generally the LAN) and the **untrusted zone** (generally the Internet). Devices that provide services to the untrusted world are generally placed in the DMZ, to provide separation from the trusted network.

A single firewall with multiple ports can be used to implement a logical DMZ:



A more secure DMZ (referred to as a **screened subnet)** utilizes multiple firewalls:

* * *