Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials: Network, Endpoint, and Cloud (Security 401)"
at http://www.giac.org/registration/gsec

Deploying a Vyatta Core Firewall

GIAC (GSEC) Gold Certification

Author: Jason Todd, jtodd@westernu.edu
Advisor: Tim Proffitt

Accepted: MMMM Dth YYYY

Abstract

This paper is intended to provide a practical installation guide for deploying a network firewall utilizing Vyatta Core 6.0, an open source network security product. Vyatta systems are intended as a lower cost replacement for Cisco routers and security appliances. This paper is ideal for small or medium sized organizations that have IT support and desire to replace or enhance existing security appliances but are faced with ever tightening budgets. Although the scope of this paper will focus on protecting a development server farm, this paper can be used as guidance for deployment at the border or other location on the network. Readers of this paper will benefit from the introduction and exploration of a lower cost, open source alternative to commercial security appliances.

© 2010 SANS Institute, Author retains full rights.

© 2010 The SANS Institute Author retains full rights.

1. Introduction

It wasn't long ago when simply placing a firewall in between your clients, servers, and the Internet was considered a good information security measure. For some organizations the firewall was put in place to simply check a box on a security audit without giving it much attention as to what the firewall was actually protecting, or not protecting. For many the firewall was viewed as a burden and tunneling protocols were developed to make applications work with firewalls.

The first firewalls were designed to work with the applications they were put in place to protect. The applications themselves were not expected to work with the firewall. (Freed, 2000) Many security administrators found that this model left their assets under protected. In an attempt to improve security, security administrators ended up closing down all but just a few of the most common ports. This practice left security administrators feeling more protected but it left many applications disconnected.

As firewalls became more locked down, application authors developed ways around firewalls. The most common way to get around a firewall is to have the application communicate over ports that are commonly left open by the firewall administrators; with TCP ports 80 and 443 being the most common. Many tunneling protocols such as SSL VPN use this technique to successfully get through firewalls. When tunneling or going over common ports isn't an option, users both innocent and malicious alike will almost always find a way through or around a firewall. It is impossible for the firewall administrator to selectively block all malicious vectors.

This cat-and-mouse game of blocking paths and finding new paths has actually led to the traditional firewall becoming ineffective at stopping threats, especially threats from within the internal network. If all we can do is block static ports then developers will just find the ports that are left open. If this happens then what have we actually prevented?

Jason Todd, jtodd@westernu.edu

© 2010 The SANS Institute Author retains full rights.

The next generation of firewalls cannot be just static packet filters. While this may prove effective for preventing unauthorized inbound connections to poorly configured servers, it is essentially ineffective at protecting applications.

The dawn of malware is still rising and malware is at or near the top of many threat lists. One of the primary methods malware gets onto systems is through vulnerable client-side software. Client-side applications are not patched as frequently as server applications. (Price 2010) Plus as more applications move off of physical hardware and into virtual environments and clouds the traditional firewall becomes even less effective.

Legacy firewalls were originally deployed at the network border. This was an ideal location since all Internet traffic would have to flow through it. As networks become more complex however there is often a need to deploy firewalls between multiple network segments.

With virtualization and cloud computing emerging, entire networks segments often times exist completely within a virtual environment. This makes a unique challenge for firewall administrators.

The next generation of firewalls needs to not only protect the physical network, they need to protect virtual environments as well. Firewalls running on virtual environments not only protect the virtual guests; they are able to protect physical networks and hosts as well. Virtual firewalls also provide a platform for testing and deploying new configurations without the need for additional hardware.

This is where Vyatta comes into play. Vyatta offers a line of licensed and unlicensed products for protecting both the physical and virtual environments. This document will discuss installation and configuration of a Vyatta Core system in a virtual environment with four separate networks. The Vyatta Core system will provide routing, firewall, and intrusion prevention.

2. An overview of Vyatta offerings

Vyatta came together in 2005 to develop an open-source alternative to traditional routers. By 2006 Vyatta released its Open Flexible Router that included a stateful packet

© 2010 The SANS Institute Author retains full rights.

inspection firewall. Today Vyatta offers three editions, Vyatta Core, Vyatta Subscription Edition, and Vyatta Plus.

2.1. Vyatta Core

Vyatta Core is the open-source version of Vyatta. It is free to download and use however there is no commercial support. Vyatta does host a community with documentation and a support forum where members and Vyatta employees assist with questions and discussions. Many of the features Vyatta offers come from the community. (Vyatta, 2010a) The Vyatta Core's main offerings are IPv4 and IPv6 routing, stateful firewall, IPSec and SSL VPN, and intrusion prevention.

It is important to realize that Vyatta Core only supports Ethernet interfaces. WAN interfaces support such as DSL, T1, or T3 require a Vyatta Subscription Edition license. For many users this limitation is not a hindrance since Ethernet delivery is becoming more and more common.

2.2. Vyatta Subscription Edition

Vyatta Subscription Edition offers everything Vyatta Core offers plus some advanced features such as: an API for centralized management, WAN device support, and TACACS+. The Vyatta Subscription Edition is commercially supported by Vyatta and is eligible for professional services consulting. Vyatta Subscription Edition also provides access to a customer only knowledge base.

It is important to note that Vyatta will not support the Subscription Edition when Vyatta Core is in use within the same organization. This means that if an organization requires enterprise support then all Vyatta installations must be Subscription Edition. (Vyatta 2010b)

2.3. Vyatta Plus

Vyatta Plus is an add-on service for Vyatta Subscription Edition. Currently the only Vyatta Plus service available is VyattaGuard web filtering. VyattaGuard is an enhancement to the web filtering offered in Vyatta Core. VyattaGuard is to further reduce web-based threats and manage bandwidth consumption. VyattaGuard claims to offer the

© 2010 The SANS Institute Author retains full rights.

most comprehensive listing of phishing, spammed, malicious, and compromised sites available. (Vyatta, n.d.)

3. Downloading and installing Vyatta Core

Obviously, great care and planning needs to be taken in account when designing and implementing a firewall. One of the most important things to consider is what assets is the firewall going to protect. This will help determine the proper placement of the firewall in the network. Part of this planning should also include written policies that the firewall will be used to enforce.

Many organizations choose to place a firewall at the border. This approach may be the least expensive and the easiest to deploy but what about attacks from the inside? A firewall cannot stop traffic that does not pass through it making a border firewall useless from preventing attacks from within the network.

Most modern operating systems include host-based firewalls. These are good at protecting threats completely within the internal network however they are dependent on the operating system. They also do not always scale well in large networks. (Deal, 2005)

Another approach is to have a firewall at the boundary of every network. This approach works well when hosts that have similar function sit on the same network. This can get very expensive using traditional firewall appliances.

Vyatta has the unique advantage to other security appliances in that it can be installed on almost any i386 hardware. This also includes installation in virtual environments. This makes it possible to put several entire networks in a virtual environment with the Vyatta providing the routing, firewall, and IPS protection. The Vyatta system offers significant cost savings whether it is installed on physical hardware or installed as a virtual machine.

3.1. Downloading

Vyatta Core is freely available for download from the web site. It is downloaded as a virtualization ready live CD ISO image. There are also virtualization templates

© 2010 The SANS Institute Author retains full rights.

available for VMware ESX and Citrix XenServer. After downloading it is a good idea to calculate the ISO's MD5 hash and compare it to the MD5 hash provided on the download page.

Once downloaded and verified the ISO image can either be burned to CD and placed in the CDROM drive or mounted directly in a virtualization environment. After loading the CD, power on the system and allow the image to boot. During the boot process do not be alarmed by IO errors for FD0 if you do not have a floppy drive present.

After the live image is booted you will need to login using "vyatta" as the username and "vyatta" as the password. Once logged in, it is possible to begin to make configuration changes however these changes will be lost at reboot unless you save them to physical media. To have a persistent configuration Vyatta must be installed to a hard disk or other bootable media.

3.2. Installing

The Vyatta live CD includes an installation script that walks through the installation process. Before installing it is advisable to save the current configuration by issuing the following command:

```
save <optional file name>
```

It is not necessary to save the configuration if no changes were made to the system from boot time to installation. The script is started by entering the following command:

```
install-system
```

The installation script will then detect the available drives. If you want to use the entire disk then just take the default options that are displayed in brackets. Otherwise the installation script allows for manual partitioning of the installation drive.

The script then asks what configuration file to copy to the installation drive. Accept the default here unless you have a previously saved configuration file in a location other that the default.

© 2010 The SANS Institute Author retains full rights.

Next the script prompts for a password for the administrator account. This is a new password for the installation and not the default password for the live CD. It is very important that a strong password is chosen here. This password will be hashed and recorded in the configuration file. Choosing a strong password will increase the time a brute force attack on the hash will take. Also, do not use "vyatta" as the administrator password. This is a well-known password.

The final thing the installation script does is install GRUB. Again, it is advisable to take the default option unless you have specific needs. Once the installation is complete the system can be restarted and booted from the installation drive. Be sure any configuration changes are saved before restarting.

4. Configuration

After the installation is complete it is time to configure the Vyatta Core system. The configuration for this example will contain four networks protected by the Vyatta system which will control the traffic between Private, Public, and Server Farm zones. The Private zone will consist of two networks. See Figure 4.1.

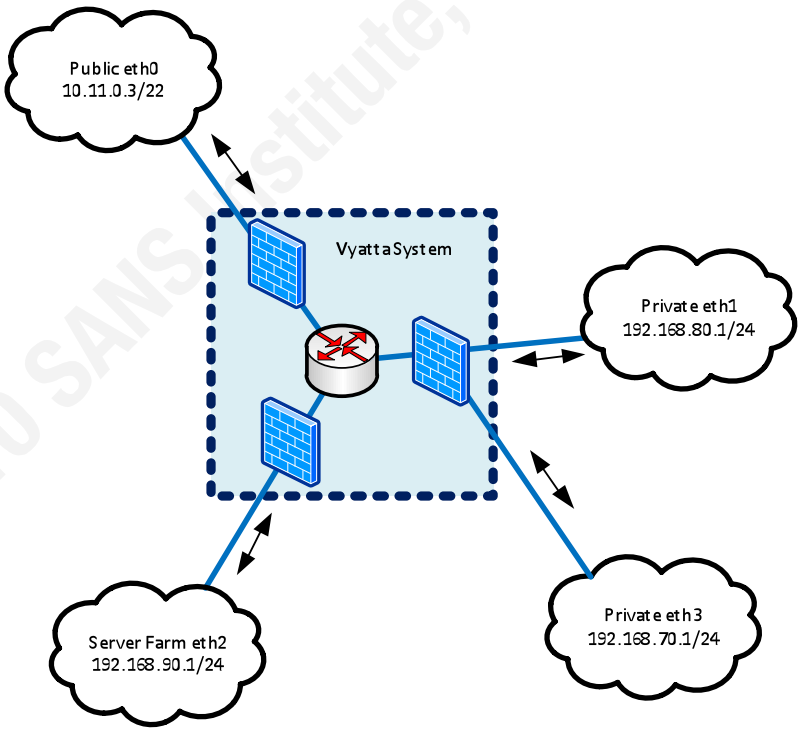

Figure 4.1 – Networks protected by the Vyatta system.

© 2010 The SANS Institute Author retains full rights.

Configuration mode is entered using the following command:

> *Configure*

4.1. Configuring the basic system

The very first thing to configure is the IP address for each of the interfaces. In configuration mode enter the following commands for to configure each interface:

> *set interfaces ethernet eth0 address 10.11.0.3/22*
> *set interfaces ethernet eth1 address 192.168.80.1/24*
> *set interfaces ethernet eth2 address 192.168.90.1/24*
> *set interfaces ethernet eth3 address 192.168.70.1/24*

Next set the default gateway address and the address of the name servers:

> *set system gateway-address 10.11.0.1*
> *set system name-server address.of.your.dns.server1*
> *set system name-server address.of.your.dns.server2*

Delete the default NTP server and set the NTP servers for your organization:

> *delete system ntp-server 0.vyatta.pool.ntp.org*
> *set system ntp-server address.of.your.ntp.server1*
> *set system ntp-server address.of.your.ntp.server2*

Commit the changes to make them active:

> *Commit*

It is important to set your NTP server address to the standard time sources for your organization and delete the default NTP server. This will make it easier to correlate the Vyatta logs with logs from other systems that are receiving time from the same NTP source. A difference of a few seconds can make it very difficult to compare the logs of separate systems.

The *commit* command is just as it sounds. It commits and activates any changes since the last commit command. It is important to understand that the *commit* command does not save the configuration. Issue the *save* command to save the configuration. The

© 2010 The SANS Institute Author retains full rights.

© 2010 SANS Institute, Author retains full rights.

commit command has the advantage of making several changes and then committing them all at once. If you run into a situation where a commit leaves the system behaving unexpectedly then all you have to do is load a previously saved good configuration file to restore to a previous state.

4.2. Configuring NAT

At this point the Vyatta system is a simple router with three Ethernet interfaces. The next step is configuring NAT to allow the Private and Server Farm networks share public IPs.

For this example we are going to NAT both our Private networks and our Server Farm. We are using NAT for both zones because we are going to take advantage of the added security NAT offers by making hosts without forwarding rules unreachable from the Internet.

NAT rules are matched in order as they are evaluated and are applied to the first match. This is important to keep in mind when configuring NAT. For example if NAT rule 10 looked for a source address in 192.168.80.0/24 and NAT rule 20 looked for the single IP 192.168.80.20 then 192.168.80.20 would match NAT rule 10 since it is part of 192.168.80.0/24 and would never match NAT rule 20. This is handled simply by setting the more specific NAT rules with lower numbers and using a high numbered catch-all for entire networks. First it is necessary to add a few more IPs to eth0 for NAT to use.

```
set interfaces ethernet eth0 address 10.11.0.101/22
set interfaces ethernet eth0 address 10.11.0.102/22
set interfaces ethernet eth0 address 10.11.0.103/22
```

Create a source NAT (SNAT) rule for our webserver:

```
set service nat rule 10 type source
set service nat rule 10 outside-address address 10.11.0.101
set service nat rule 10 source address 192.168.90.100
set service nat rule 10 outbound-interface eth0
```

Create a destination NAT (DNAT) rule so our webserver is reachable from the Internet:

© 2010 The SANS Institute Author retains full rights.

```
set service nat rule 20 type destination
set service nat rule 20 destination address 10.11.0.101
set service nat rule 20 inbound-interface eth0
set service nat rule 20 inside-address address 192.168.90.100
```

Create a source NAT (SNAT) rule for our SMTP gateway:

```
set service nat rule 11 type source
set service nat rule 11 outside-address address 10.11.0.102
set service nat rule 11 source address 192.168.90.20
set service nat rule 11 outbound-interface eth0
```

Create a destination NAT (DNAT) rule so our SMTP gateway is reachable from the Internet:

```
set service nat rule 21 type destination
set service nat rule 21 destination address 10.11.0.102
set service nat rule 21 inbound-interface eth0
set service nat rule 21 inside-address address 192.168.90.20
```

Create a catch-all SNAT rule for our Private zone clients and commit the changes

```
set service nat rule 1000 type source
set service nat rule 1000 outbound-interface eth0
set service nat rule 1000 outside-address address 10.11.0.103
set service nat rule 1000 source address 192.168.70.2-192.168.80.254
commit
```

After configuring NAT it is a good idea to verify that it is working properly. This can be done by issuing the following commands outside of configure mode:

```
show nat rules
show nat translations
```

The NAT translations will not be displayed until a session is started. It is also a good idea to verify on a remote system that the NAT rules are working properly. This can be done by viewing log the files or running a protocol analyzer such as tcpdump on the remote host. See figure 4.2.

© 2010 The SANS Institute Author retains full rights.

```
[jason@localhost ~]$ sudo /usr/sbin/tcpdump -nnt -i seth0 host 10.11.0.103
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on seth0, link-type EN10MB (Ethernet), capture size 96 bytes
IP 10.11.0.103.1291 > 10.7.250.191.80: S 2492543691:2492543691(0) win 65535 <mss 1460,nop,nop,sackOK>
IP 10.7.250.191.80 > 10.11.0.103.1291: S 2328452358:2328452358(0) ack 2492543692 win 5840 <mss 1460,nop,nop,sackOK>
IP 10.11.0.103.1291 > 10.7.250.191.80: . ack 1 win 65535
IP 10.11.0.103.1291 > 10.7.250.191.80: P 1:488(487) ack 1 win 65535
IP 10.7.250.191.80 > 10.11.0.103.1291: . ack 488 win 6432
IP 10.7.250.191.80 > 10.11.0.103.1291: P 1:145(144) ack 488 win 6432
IP 10.7.250.191.80 > 10.11.0.103.1291: F 145:145(0) ack 488 win 6432
IP 10.11.0.103.1291 > 10.7.250.191.80: . ack 146 win 65391
IP 10.11.0.103.1291 > 10.7.250.191.80: F 488:488(0) ack 146 win 65391
IP 10.7.250.191.80 > 10.11.0.103.1291: . ack 489 win 6432
```

Figure 4.2 – tcpdump output on a webserver showing NAT address.

4.3. Configuring the firewall

The firewall can be configured after NAT is configured. Like NAT, the firewall rules are matched in the order they are evaluated so it is important that more specific rules get lower numbers than more general rules.

The Vyatta firewall can be applied to either interfaces or zones. A zone can contain one or more interfaces. Zones are useful for grouping networks and controlling flow from one group of networks to another. Each interface in a zone has the same security level.

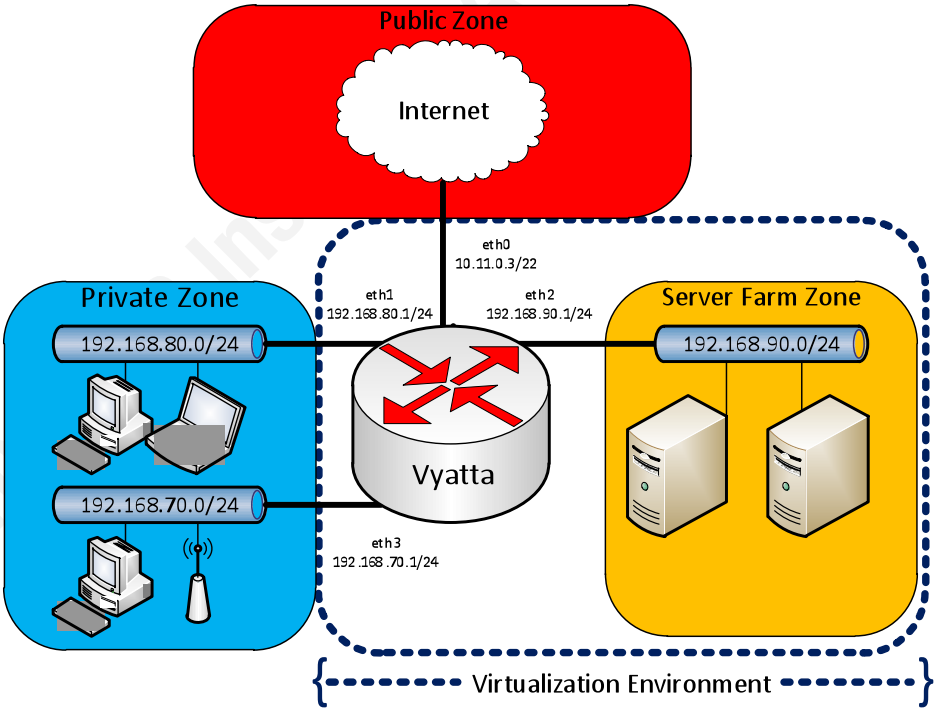

Figure 4.3 – Firewall zones. Adapted from *Firewall reference guide* (p. 23), 2010, Vyatta, Inc.

Jason Todd, jtodd@westernu.edu

© 2010 The SANS Institute Author retains full rights.

For example, figure 4.3 shows the Vyatta firewall with zones defined for Private, Public, and Server Farm. The interfaces eth1 and eth3 are in the Private zone. Any firewall rules applied to the Private zone will be applied to traffic on these interfaces. Traffic flowing from interfaces within the same zone is not filtered. Also, traffic flowing between interfaces not assigned to a zone will flow unfiltered.

The Vyatta firewall uses a default deny model so traffic that is flowing between zones that is not explicitly configured is dropped. The first thing to do then is to create some rules to allow traffic. The zone-based firewall is configured in two steps. The first step is to define the firewall rule sets and the second step is to apply the firewall rules sets to each zone and direction. This is necessary because firewall rules for zones are unidirectional.

The most flexible configuration would require a pair of firewalls defined for each zone that would need to pass traffic to one another. This way the firewall would have different rules for filtering incoming outgoing traffic.

This would mean that a Vyatta system with three zones would need six firewalls configured. Twenty-four firewalls would be required for four zones and forty firewalls would need to be configured for a firewall with five zones. The formula is 2N(N-1) where N is the number of firewall zones.

This obviously gets to be impractical as the number of zones increases. To make management more practical, firewalls can be created to control the flow of traffic to each zone without taking into account where the traffic originated. This cuts the number of firewalls configured on the Vyatta in half. It is possible to further reduce the number of firewalls configured on the Vyatta by applying the same firewall to multiple zones. This may not be desirable however because it requires more generalizations which in turn leads to less security and less flexibility.

When defining firewall rules for zones it is useful to name the firewall rules according to flow direction and the zones the firewall rules will be applied to. Firewall rules that apply to traffic egress from the Vyatta to the Public zone can be named to-public. Likewise, firewall rules that apply to traffic egress from the Public zone to the Server Farm zone can be named public-to-serverfarm.

Jason Todd, jtodd@westernu.edu

© 2010 The SANS Institute Author retains full rights.

In this example we will configure four zones: to-public, to-private, public-to-serverfarm, and private-to-serverfarm. This will allow the Server Farm to be reachable by different policies for both internal, Private zone users and external, Public zone users. Having different policies will allow management activity, such as SSH from the Private zone but block it from the Public zone.

The Server Farm in this example will also not be able to successfully start a session with any client outside of the Server Farm zone. The three-way handshake will fail when the remote client replies with a SYN-ACK. This is because the firewalls for the Server Farm zone will be configured to only accept connections on allowed TCP ports and they will not be configured to allow established or related sessions.

Clients from the Public and Private zones will however still be able to establish sessions with the Server Farm zone. This is controlled by allowing established and related sessions with the Public and Private zones but not with the Server Farm Zone. See figures 4.3 and 4.4

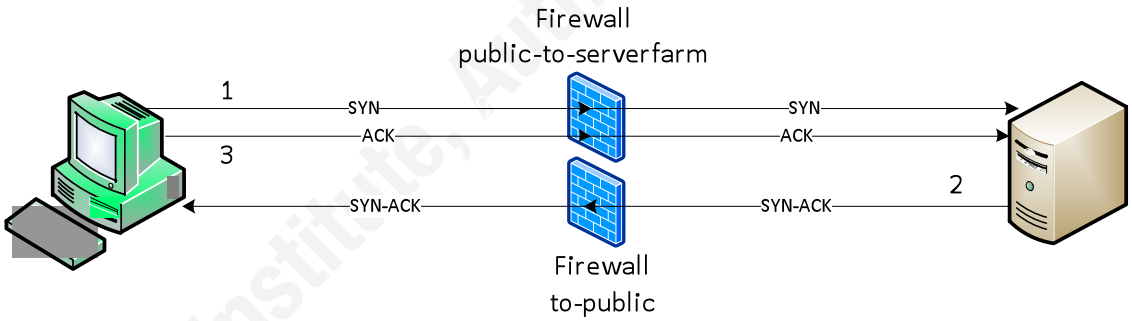

Figure 4.3 – Complete TCP three-way handshake when session is initiated from a client.

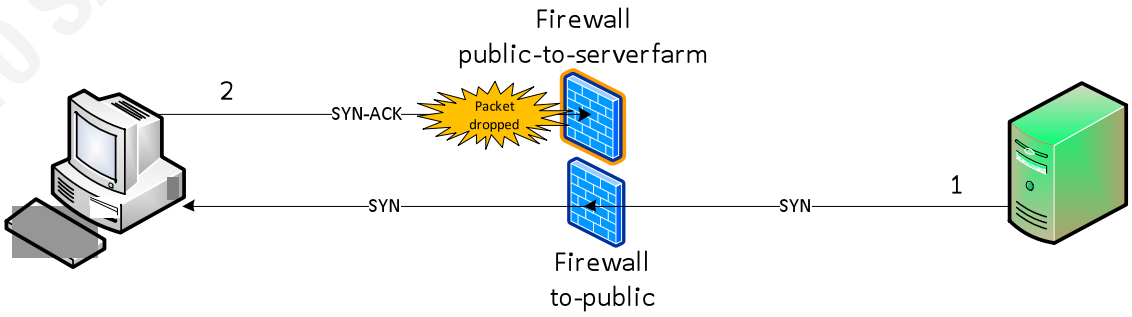

Figure 4.4 – Failed TCP three-way handshake when session is initiated from a server.

Jason Todd, jtodd@westernu.edu

© 2010 The SANS Institute Author retains full rights.

This security configuration assumes that the servers in the Server Farm zone will only respond to client connections. The servers will never need to start a session of their own. If a server compromise is attempted, the server will be prevented from establishing a connection back to the attacker. While this alone may not block an attack, it will help reduce the chances of successful control of a compromised server.

Zones are created by issuing the following commands in configure mode:

```
set zone-policy zone private description "Private zone (inside firewall)"
set zone-policy zone public description "Public zone (outside firewall)"
set zone-policy zone serverfarm description "Server Farm zone"
```

Next the zones are assigned to interfaces:

```
set zone-policy zone public interface eth0
set zone-policy zone private interface eth1
set zone-policy zone serverfarm interface eth2
set zone-policy zone private interface eth3
Commit
```

At this point no traffic will be able to flow between the zones since all inter-zone traffic is dropped. To allow traffic to flow we first need to configure the firewall rule sets then apply the newly created rules to zones.

First, a firewall rule set needs to be created for the Public Zone. In this example only TCP ports 80 and 443 are allowed to any host and UDP 53 is allowed to the authorized DNS servers:

```
set firewall name to-public description "Allow authorized traffic to Public Zone"
set firewall name to-public rule 1 action accept
set firewall name to-public rule 1 state established enable
set firewall name to-public rule 1 state related enable
set firewall name to-public rule 10 action accept
set firewall name to-public rule 10 destination port 80,443
set firewall name to-public rule 10 protocol tcp
```

Jason Todd, jtodd@westernu.edu

© 2010 The SANS Institute Author retains full rights.

```
set firewall name to-public rule 20 action accept
set firewall name to-public rule 20 destination address 10.11.2.49-10.11.2.50
set firewall name to-public rule 20 destination port 53
set firewall name to-public rule 20 protocol udp
Commit
```

Next a firewall rule set needs to be created for the Private Zone. In this example only established connections are allowed:

```
set firewall name to-private description "Allow established to Private zone"
set firewall name to-private rule 1 action accept
set firewall name to-private rule 1 state established enable
set firewall name to-private rule 1 state related enable
Commit
```

The third firewall rule set is created to allow traffic from the Public zone to the Server Farm zone. In this example TCP ports 80 and 443 are allowed to the web server and TCP port 25 is allowed to the SMTP gateway:

```
set firewall name public-to-serverfarm description "Public to Server Farm"
set firewall name public-to-serverfarm rule 10 action accept
set firewall name public-to-serverfarm rule 10 destination port 80,443
set firewall name public-to-serverfarm rule 10 protocol tcp
set firewall name public-to-serverfarm rule 10 destination address 192.168.90.100
set firewall name public-to-serverfarm rule 20 action accept
set firewall name public-to-serverfarm rule 20 destination port 25
set firewall name public-to-serverfarm rule 20 protocol tcp
set firewall name public-to-serverfarm rule 20 destination address 192.168.90.20
Commit
```

It may also be a good idea to limit the number of connections by any one client from the Public Zone to the Server Farm zone. Limiting the connections could help slow a brute force or a DOS attack. This is done by creating a rule that limits number of connections for a period of time:

```
set firewall name public-to-serverfarm rule 5 action drop
set firewall name public-to-serverfarm rule 5 protocol tcp
set firewall name public-to-serverfarm rule 5 destination port 25,80,443
```

Jason Todd, jtodd@westernu.edu

© 2010 The SANS Institute Author retains full rights.

```
set firewall name public-to-serverfarm rule 5 recent count 20
set firewall name public-to-serverfarm rule 5 recent time 5
Commit
```

These settings limit the number of connections to the Server Farm on TCP ports 25, 80 and 443 to 20 in 5 seconds. Additional rules can be added to protect other services with different thresholds.

The final firewall rule set is to allow traffic from the Private zone to the Server Farm zone. In this example TCP ports 21, 80, 443 are allowed by any host in the Private zone. The webmaster is allowed to connect to port TCP port 22 on webserver and the postmaster is allowed to connect to port 22 and 25 on the SMTP gateway for management:

```
set firewall name private-to-serverfarm description "Private to Server Farm"
set firewall name private-to-serverfarm rule 1 action accept
set firewall name private-to-serverfarm rule 1 state established enable
set firewall name private-to-serverfarm rule 1 state related enable
set firewall name private-to-serverfarm rule 10 action accept
set firewall name private-to-serverfarm rule 10 destination port 21,80,443
set firewall name private-to-serverfarm rule 10 protocol tcp
set firewall name private-to-serverfarm rule 20 action accept
set firewall name private-to-serverfarm rule 20 destination port 22
set firewall name private-to-serverfarm rule 20 protocol tcp
set firewall name private-to-serverfarm rule 20 destination address 192.168.90.100
set firewall name private-to-serverfarm rule 20 source address 192.168.80.10
set firewall name private-to-serverfarm rule 30 action accept
set firewall name private-to-serverfarm rule 30 destination port 22,25
set firewall name private-to-serverfarm rule 30 protocol tcp
set firewall name private-to-serverfarm rule 30 destination address 192.168.90.20
set firewall name private-to-serverfarm rule 30 source address 192.168.80.20
Commit
```

When configuring a firewall, an important thing to take into consideration is the order in which rules are applied where they are applied within the Vyatta system. If order is not taken into account then unintended results will occur. These can be obvious such as desired traffic being blocked. However, worse than that is

Jason Todd, jtodd@westernu.edu

© 2010 The SANS Institute Author retains full rights.

unintended traffic that passes through unchecked. Within the Vyatta system, the firewall sits in between destination NAT and source NAT.

Like NAT rules, the firewall rules are applied in order according to the rule number. If traffic matches a firewall rule, that action is applied to the traffic. If the traffic does not match the rule then it is analyzed by the next rule and so on until either a match is made or all of the firewall rules are exhausted. If the traffic does not match any rule then it is dropped by the Vyatta system.

The final step for configuring the firewall is assigning the previously configured firewall rule sets to the previously configured zones. This is done by assigning a firewall rule set to a direction of flow to and from each zone.

```
set zone-policy zone serverfarm from private firewall name private-to-serverfarm
set zone-policy zone serverfarm from public firewall name public-to-serverfarm
set zone-policy zone private from public firewall name to-private
set zone-policy zone private from serverfarm firewall name to-private
set zone-policy zone public from private firewall name to-public
set zone-policy zone public from serverfarm firewall name to-public
Commit
```

4.4. Configuring the IPS

The Vyatta system runs Snort for the detection engine in its Intrusion Protection System. This adds a significant advantage to simple packet filtering firewalls . Packet filtering firewalls are good at blocking packets that are destined for unauthorized services such as blocking all traffic to TCP port 21 on the mail server.

Packet filtering firewalls however are very bad at discovering packets with bad intent (Avolio, 1999) since packet filtering only looks at the source address, destination address, protocol, and port numbers. A simple packet inspection firewall would have no effect on a packet with a malicious payload destined for an authorized service. A simple packet filtering firewall would see that the packet was

© 2010 The SANS Institute Author retains full rights.

destined for an authorized service, such as port 25 on the mail server and allow it through.

A significant improvement to packet filtering is intrusion prevention which runs each packet through its content matching engine. This can be configured to drop, alert, pass, or silently drop on any packet that matches a Snort rule. Vyatta makes configuring the IPS very easy.

Since the IPS is Snort, it is necessary to configure the firewall action based on packets matching Snort priorities. Just like other Vyatta modules, the IPS is simple to enable and configure. The very first thing to do is set the action for each priority level one through three and other.

```
set content-inspection ips actions priority-1 drop
set content-inspection ips actions priority-2 alert
set content-inspection ips actions priority-3 alert
set content-inspection ips actions other pass
```

The next step is to set the auto update for the Snort rules. The auto updates require registration with snort.org in order to obtain an "oink code" to get the updates.

```
set content-inspection ips auto-update oink-code <your oink code>
```

Next the traffic that is processed by the IPS is defined. It is possible to only apply the content filter to custom firewall rule sets or simply process all traffic through the IPS:

```
set content-inspection traffic-filter preset all
```

The final step is to configure the time of day to get the updates. This is the hour, 0-23 when the Vyatta system downloads and installs the updates.

Jason Todd, jtodd@westernu.edu

© 2010 The SANS Institute Author retains full rights.

© 2010 SANS Institute, Author retains full rights.

set content-inspection ips auto-update update-hour <0-23>

This time should be carefully chosen because the Snort engine will need to restart after the new rule set is downloaded. While the Snort engine is restarting, the Vyatta system will not process traffic for five to ten seconds.

Custom Snort rules can also be written to match specific traffic for the environment being protected. For example a Snort rule can be written to detect, filter and log if a host in the Private or Server Farm Zones attempts to download a Windows dynamic-link library.

The Snot log can be shown by issuing the following command outside of configuration mode:

show ips log

```
------------------------------------------------------------------------------
2010-09-04 11:31:26.346356 {ICMP} 10.11.0.10 -> 10.11.0.103
(misc-activity) Misc activity (priority 3)
[1:384:5] ICMP PING
------------------------------------------------------------------------------
2010-09-04 11:31:27.342264 {ICMP} 10.11.0.10 -> 10.11.0.103
(misc-activity) Misc activity (priority 3)
[1:382:7] ICMP PING Windows
------------------------------------------------------------------------------
2010-09-04 11:31:27.342264 {ICMP} 10.11.0.10 -> 10.11.0.103
(misc-activity) Misc activity (priority 3)
[1:384:5] ICMP PING
------------------------------------------------------------------------------
```

Figure 4.5 – IPS log after a ping from a host in the Public Zone.

4.4.1. IPv6

Throughout this document only IPv4 has been discussed. It is however prudent to mention the importance of a using a firewall with IPv6. NAT alone is not a security measure because NAT is not intended to provide security. The security NAT offers is simply a side effect of how NAT works. There are many cases however where simply placing hosts behind NAT offers significant protection because hosts behind NAT are not directly reachable without forwarding rules.

Unlike IPv4, IPv6 does not offer any NAT functionality. IPv6 hosts are left directly exposed to untrusted networks. This makes a firewall especially important in protecting an IPv6 network. Vyatta has a full feature IPv6 firewall. The IPv6 firewall is

© 2010 The SANS Institute Author retains full rights.

separate from the IPv4 firewall meaning that rules from one firewall do not apply to the other. If your environment supports both IPv4 and IPv6 then it will be necessary to configure a firewall for each IP protocol for each zone or interface.

4.5. Configuring the Web Filter

The Vyatta system can be configured to manage internal user activity by acting as a web proxy server. The web filter can reduce exposure to web-based threats, block objectionable content, increase productivity and reduce bandwidth usage by caching content. Threats are blocked by URL filtering against a community updated category list.

Web filtering rules can block specific URLs, keywords in URLs, website category, or mime type. This makes filtering for sites with many IPs or IPs that change frequently easier since the filter is based off of the URL and not the IP address. Rules can be applied to all users, individual users, or groups of users based off IP address or subnet. Like many of Vyatta's counterparts, the Vyatta cannot establish rules based on external authentication such as Active Directory. This makes filtering for single users more difficult in dynamic and mobile environments since the user's IP address may change frequently.

The following configuration blocks the category "spyware" and "filehosting" for all users except the security administrator who has a static IP address:

```
set service webproxy listen address 192.168.80.254
set service webproxy url-filtering squidguard source-group SecAdmin /
    address  192.168.80.42
set service webproxy url-filtering squidguard source-group Users /
    address  192.168.80.0/24
set service webproxy url-filtering squidguard rule 10 source group SecAdmin
set service webproxy url-filtering squidguard rule 20 source group Users
set service webproxy url-filtering squidguard rule 20 block-category spyware
set service webproxy url-filtering squidguard rule 20 block-category filehosting
Commit
```

Just like other Vyatta configurations, web filtering rules are analyzed in order and applied to the first match. In this example the security administrator's IP address matched rule 10. The security administrator would not have any web filter

© 2010 The SANS Institute Author retains full rights.

rules applied since there are no additional filters are defined for rule 10. All other users in the subnet for the Private zone would have URLs blocked since they would match rule 20.

5. Conclusion

The Vyatta Core system is a powerful and robust alternative to more expensive network security appliances. What Vyatta has actually provided is a much needed wrapper for many open source security products. This gives security administrators the ability to configure and manage the system with a simple to use CLI and a monolithic configuration file.

Since Vyatta is as confortable in a virtual environment as being installed on physical hardware, it offers further savings from lower power consumption, lower cooling costs, and less rack space for the ever more cost conscious security administrator.

There is no silver bullet for stopping threats. Stateful packet inspection firewalls have played an important role in protecting systems. While they are still effective at limiting unauthorized access to protected systems, they are ineffective at blocking payloads with bad intent.

Vyatta has address this limitation to an extent with the inclusion of IPS and web URL filtering. The IPS can detect and block malicious payloads from untrusted networks and the URL filter can be used to control internal user web activity. The IPS can also be used to control user activity such as detecting when credit card information is being transmitted in clear text. Also, since Vyatta is open source it may be possible to build in additional application filtering using another application such as l7-filter.

Whether Vyatta is installed at the core, border, or between subnets in a virtual environment it will complement any defense in depth strategy.

Jason Todd, jtodd@westernu.edu

© 2010 The SANS Institute Author retains full rights.

6. References

Freed, N. (2000). Behavior of and requirements for internet firewalls. Retrieved (2010, August 10) from http://www.ietf.org/rfc/rfc2979.txt

Prince, K. (2010, January 29). Top 10 information security threats of 2010. *Network Security Edge*, Retrieved from http://www.networksecurityedge.com/content/top-10-information-security-threats-2010

Vyatta. (2010a). *Vyatta roadmap*. Retrieved (2010, August 10) from http://www.vyatta.org/documentation/product-roadmap

Vyatta. (2010b). *Vyatta firewall & subscription.* Retrieved (2010, September 2) from http://www.vyatta.org/forum/viewtopic.php?p=40992

Vyatta. (2010c). *Firewall reference guide.* Retrieved (2010, September 8) from http://www.vyatta.com/downloads/documentation/VC6.0/Vyatta_Firewall_R6.0_v03.pdf

Vyatta. (n.d.). *VyattaGuard web filtering*. Retrieved (2010, August 10) from http://www.vyatta.com/downloads/datasheets/vyattasfvrt.pdf

Deal, R.A. (2005). *Cisco router firewall security*. Indianapolis: Cisco Press.

Avolio, F. (1999). Firewalls and internet security, the second hundred (Internet) years. *The Internet Protocol Journal*, *2*(2), Retrieved from http://www.cisco.com/web/about/ac123/ac147/ac174/ac200/about_cisco_ipj_archive_article09186a00800c85ae.html

7. Tools mentioned in this document

Vyatta Core
http://www.vyatta.org/

tcpdump
http://www.tcpdump.org/

Snort
http://www.snort.org/

l7-filter
http://l7-filter.sourceforge.net/

Jason Todd, jtodd@westernu.edu

© 2010 The SANS Institute Author retains full rights.