Defending Against DDoS Attacks Using Open Source Software

Rodrigo Delgado Aguilar

A Thesis submitted in the partial fulfillment
of the requirements for the degree of
Bachelor of Computer Science

Algoma University College

Thesis supervised by
Prof. Gerry Davies
Department of Math and Computer Science

Second Reader:
Dr. George Townsend

Sault Ste. Marie, Ontario, CA. April 2008

Acknowledgments

For my family and friends, especially Ing. Alfonso Martinez de Castro, for giving me their unconditional support.

Abstract

Distributed denial-of-service (DDoS) attacks represent a major security problem for every internet user. A defense system against a DDoS attack should be able to detect these attacks and quickly respond in order to stop the flooding of the victim network. A DoS attack can also be created by the high demand of users in a popular website. This is why it is equally important to recognize the legitimate traffic and keep providing the service to these users.

For a DoS attack we can notice the one-to-one relationship between the attacker and the victim, therefore it might not be necessary to have any extra help since the situation depends only on two people. It is a win-lose situation. In the case of a DDoS attack the relationship clearly has an advantage for the attacker: an N-to-one relationship, where N is the number of attackers and one is the victim. In this scenario there is no win-lose situation for the victim, hence it raises the need for a Distributed Defense System against DDoS attacks.

Current solutions are only affordable by big companies or people who have thousands of dollars to spend monthly for such protection [Table 1].

The proposed defense against DDoS attacks in this document is open source based, because it is intended to be an affordable defense for every internet user who requires to be protected.

Every internet user should have the right to be protected against both DoS and DDoS attacks, especially when the internet has become a high medium of communication.

Contents

Abbreviations

APF	Advanced Policy Firewall is an iptables-based firewall system designed around the essential needs of today's Internet deployed servers and the unique needs of custom deployed Linux installations.
DoS	Denial-of-Service is an incident in which a user or organization is deprived of the services of a resource they would normally expect to have.
DDoS	A Distributed Denial-of-Service is when there is a large number of compromised systems (sometimes called a botnet) that attacks a single target.
HTTP	Hyper Text Transfer Protocol
IDS	An Intrusion Detection System generally detects unwanted manipulations of computer systems, mainly through the Internet.
IP	Internet Protocol.
IPS	An Intrusion Prevention System monitors network traffic. However, because an exploit may be carried out very quickly after the attacker gains access, IPS also have the ability to take immediate action, based on a set of rules established by the network administrator.
IRC	Internet Relay Chat is a form of real-time Internet chat or synchronous conferencing.
LAN	A Local Area Network is a group of computers and associated devices that share a common communications line or wireless link.
MRTG	Multi Router Traffic Grapher. Monitors SNMP network devices and draws pictures showing how much traffic has passed through each interface.
NETSTAT	Network Statistics.
OS	Operating System.
PHP	The PHP Hypertext Preprocessor allows web developers to create dynamic content that interacts with databases.
SMTP	Simple Mail Transfer Protocol.
SSH	Secure Shell.
SOHO	Small Office/Home Office.
URL	Uniform Resource Locator.
ZmbScap	Zombie Scapper is an automated Perl tool for detecting and stopping distributed denial of service programs.

List of Tables

List of Figures

1. Introduction

The internet has become a very important, if not the most important, media of communication for people and business. As Internet usage grows so will its problems.

Viruses and Worms are the most common Internet problems [1]. Because of this, the creator of such malware can take advantage of infected computers to do whatever he wants with them. One example is a Distributed Denial-of-Service attack [2].

We will now discuss what a Denial-of-Service Attack is and its types.

1.1 Denial-of-Service Attacks

A DoS attack deprives internet users of the services of a resource they would normally expect to have. Some examples include:

- Flooding a network preventing legitimate network traffic.
- Consumption of network resources such as bandwidth, memory, CPU, and disk space causing a network slowdown or even to bring it down.
- Disrupting communication between two computers preventing access to a service including email and databases.
- Disrupting a service from a specific system or person.

DoS attacks can result in significant loss of time and money for many organizations [3]. Not all DoS attacks are intentional; a website could end up denied, not because of an attack but because of its extreme popularity. If thousands of people simultaneously visit the website for a few hours then it will be the same effect as in a DDoS attack.

An analogy for understanding better a DoS attack is the "pizza prank call":

"Imagine a hacker creates a program that calls a local pizza store. The pizza store answers the telephone, but learns that it is a prank call. If the program repeats this task continuously, it prevents legitimate customers from ordering pizza because the telephone line is busy. This is a denial of service, and analogous to a DoS attack.[4]"

Now we will discuss in a more technical view how a DoS and DDoS attack works.

1.1.1 DoS

The simplest yet harmful DoS attack. Here an attacker attempts to disrupt the communication between client and servers to prevent them from accessing information or services such as email, web sites, banking, or any other services that rely on the affected computer.

The most common and obvious type of DoS attack occurs when an attacker floods a network [5]. Because the server can only process a certain number of requests at once, if an attacker overloads the server with

requests then the server will not be able to process your request. This is a denial-of-service because you can not access to the service you would expect to have.

Some common DoS attacks are ICMP floods, Teardrop attacks, Peer-to-Peer attacks, IRC attacks, and Nuke attacks. All of these attacks overload the victim's resources making them unavailable for others.

1.1.2 DDoS

This attack consists of a network where the attacker has control over infected computers (zombies) dispersed worldwide that attack a single target. Such a network is sometimes denominated botNET. The attack is distributed because the attacker is using multiple computers to launch the denial-of-service attack (Figure 1).

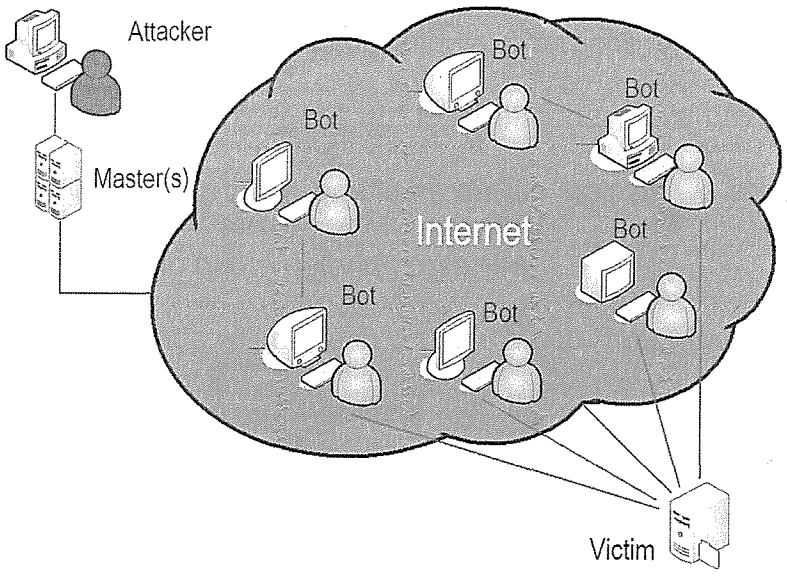


Figure 1. A DDoS Architecture

A botNET can be controlled either via IRC or from a Website. Both of these methods produce the same damage. They just differ in how the attacker manipulates the zombies in order to attack his victim.

In a case where a DDoS is integrated with thousands of zombies it can be used to try to bring down the Internet worldwide affecting everyone who uses it [6].

1.2 Understanding the Attacker

In order to explain how DoS attacks can be done, we need to understand why these attacks are executed. First we need to know that an attacker can be anyone who has access to the internet. Second we have to discard the idea that attackers are just Hackers. As a matter of fact the term you would use here is 'Cracker'. A Cracker is someone who breaks into someone else's computer system for profit, maliciously, for some altruistic purpose or cause, or because the challenge is there [7]. Finally but not least important, we have to know that there are legal solutions for these problems but as long as we are under attack there is nothing we can do but to defend ourselves.

The extent of damage will depend on how much "hate" the attacker has for you. In example, a DDoS attack that lasts only for one day is more likely to be just for fun rather than one which lasts for weeks. The number of zombies in a botNET can also determine this factor.

1.3 Attacking Methods

There are two methods often used for DDoS attacks: The first one involves infected computers connected to an IRC channel, this method is known as IRC-based botNET; and the second one is a web-based botNET. Both of these methods can produce the same damage to the victim's network.

We will now discuss both of these methods used in DDoS attacks.

1.3.1 IRC-based botNET

An IRC-based attack is a network of zombies controlled by a master zombie computer (Figure 1). All zombies are connected to an IRC channel. The master zombie will tell its zombies when to attack a specific target.

Usually the users of such zombie computers are not aware that they are being part of a DDoS attack [8].

1.3.2 Web-based botNET

This method is similar to an IRC-based botNET except that in here the zombies are controlled by the attacker using a website. Via this website the attacker can send commands and manipulate the zombies against their will (See Appendix A).

2. Technical Background

DDoS attacks can be used for extortion, market competition, political sabotage or even cyber terrorism [9]. This means everyone is susceptible to a DDoS attack even if the attack has no profit-purpose. An example of such situation will now be discussed.

2.1 Real Life Incidents

Last year one of the biggest Spanish-spoken websites regarding hacking, elhacker.net, suffered a DDoS attack that caused the site to be down for month and a half.

The attack consisted of 40,000 compromised computers which flooded the server day and night. The webmaster had a talk with the attacker and he find out that the attacker was doing it just because he felt that elhacker.net was full of "lamers". Lamer is used to describe someone who is intentionally ignorant of how things work.

The site elhacker.net has been attacked numerous times and it is scanned everyday hundreds of times to find any vulnerability in the site's code or security. Apparently all of these attacks are just because of the site name: "elhacker" which means "The Hacker". Attackers just want to prove how much the webmaster knows about hacking, which implies security, regardless of not gaining any profit for doing it [10].

2.2 Current Solutions

Current DDoS solutions consist of implementing an IPS [11], subscribing to an Anti-DDoS service from the Internet Service Provider [12], and/or to implement a special hardware device in your network [13]. All these solutions implement packet-filtering to prevent malicious data from reaching the victim's network and causing a DoS. An estimate for these services is shown in next table:

Company	Type of Service	Estimated Price (USD)
Prolexic	Basic	$8000 + setup
RioRey	Basic	$9,500
IIJ (Japan)	Basic	$6,400

Table 1. Comparing Anti-DDoS Service Prices.

As we can see from above, such prices are not accessible for everyone. All these services are more company-focused rather than SOHO users.

2.3 Proposed Solution

Since there are billions of internet users worldwide (Figure 2), with Microsoft Windows the most used operating system [14] and the operating system with most vulnerabilities (in operating system itself, not with 3rd-party

components) [15], indicates indeed that there is a high risk for an internet user to be attacked by a DDoS. A need for a "free" defense arises from these factors.

As we saw before, current solutions are only for people who can spend thousands of dollars for such protection, or for companies that do not want to be victims of a DoS attack and lose profits because of this. These solutions, as stated before, are methods that use packet-filtering to prevent malicious data from disrupting the service between server and client. A simple internet user can achieve a solution for this attack at an affordable price: time and effort.

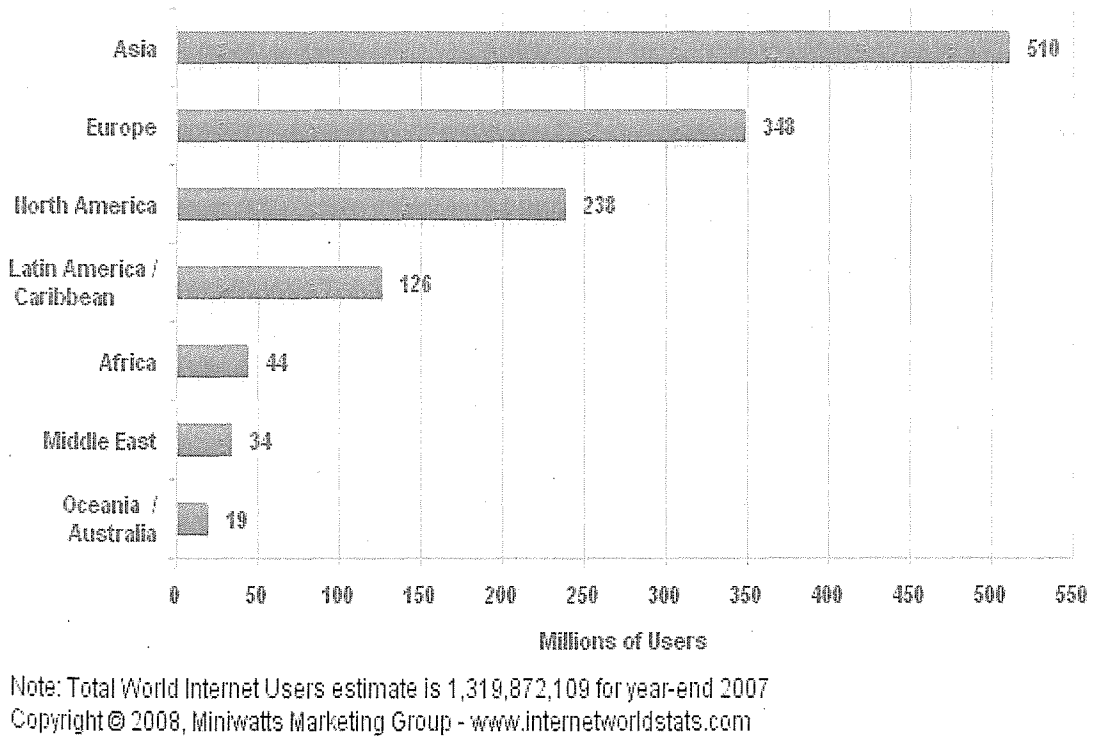


Figure 2. Internet Users in the World - December 2007

3. Experimental Environment

To create a Distributed Defense against DoS attacks, without the need of spending thousands of dollars monthly, we will apply some methods used by current solutions and the attackers in a SOHO network (Figure 3).

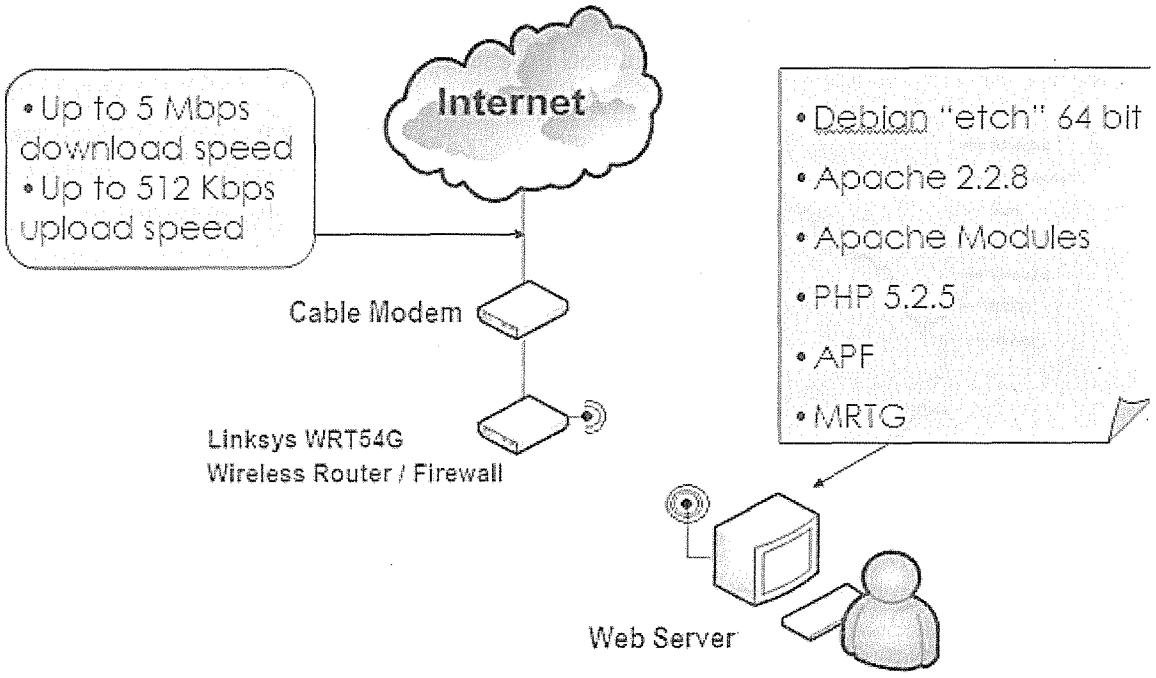
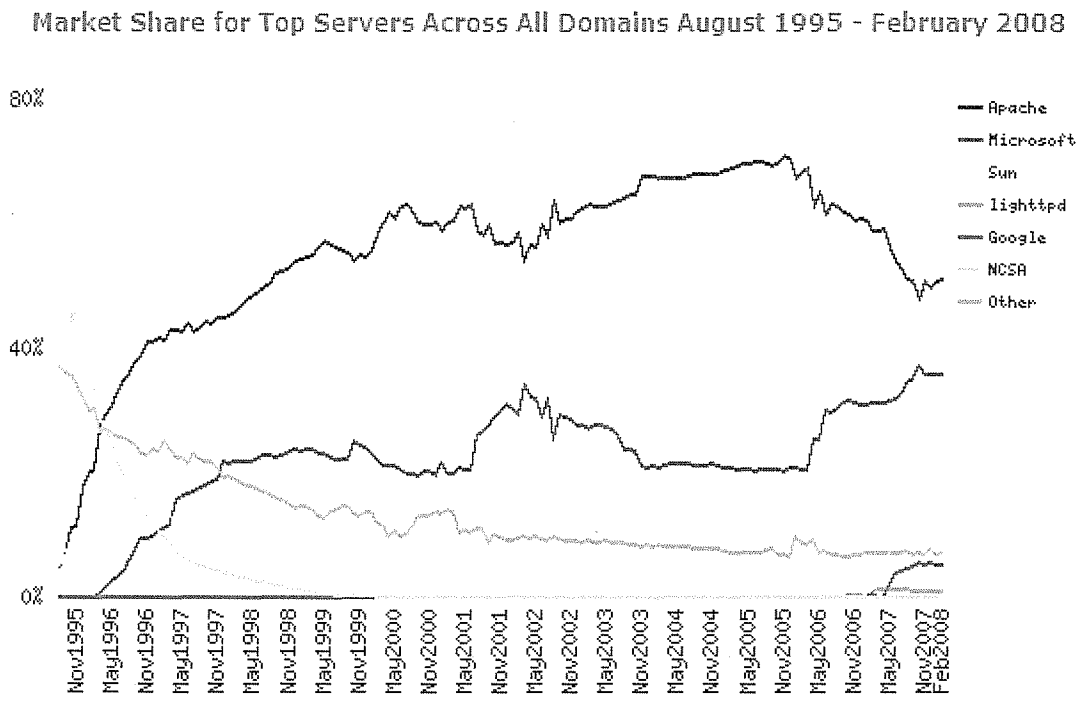


Figure 3. The SOHO Experimental Network

This paper assumes, based on research (see Appendix B), that a victim is commonly a website. Therefore the experimental environment will be tested using Apache web server mounted on a Linux operating system. The reason is because Apache is the most used web server (Figure 4) and in conjunction with Linux it offers a wide variety of customizable add-ons to protect and serve a

website more efficiently. Another important advantage is that all the software used in this experiment is Open Source. This means that the protection will be affordable for everyone who mounts a web server using Linux. We need to emphasize that this defense will not be "free" at all because we are still paying for services such electricity, internet service, etc., other than that, the software is free for use.



Top Developers

Developer	January 2008	Percent	February 2008	Percent	Change
Apache	78,735,581	50.61%	80,580,183	50.93%	0.33
Microsoft	55,709,926	35.81%	56,265,527	35.56%	-0.24
Google	8,290,471	5.33%	8,169,930	5.16%	-0.16
lighttpd	1,536,981	0.99%	1,565,536	0.99%	0.00
Sun	557,673	0.36%	547,510	0.35%	-0.01

Figure 4. Market Share for Top Servers

3.1 The Project Network and its Components

To ensure performance, stability, and full customization of our web server [16] the following software is being used for this experimental Distributed Defense:

- Debian 4.0 r3 "etch" 64bit – Operating System
- Apache 2.2.8 – HTTP server.
- Apache Modules – Modularizes web server functionality.
- PHP 5.2.5 - Scripting language for web development.
- MRTG - Monitors SNMP network devices.
- APF - An iptables-based firewall system.
- SSH – Allows data to be exchanged between two computers.
- ZmbScap – Detects and stops DDoS programs.

Some of these components will be configured according to our server needs, others will be left with their default configuration.

3.1.1 Apache Web Server

Apache Web Server for Linux is very customizable at all levels. You don't need to be an expert to modify the configuration file of Apache to make it a better

Web Server, but of course you need to know what you wish to modify and the risks of doing it.

For this experiment, since a DoS attack is commonly caused by flooding and consuming our system resources, the Apache configuration file is modified so it can handle more client connections, reduce timeout limits, and add modules to help improve its security.

The server will be less susceptible to DoS attacks but it will consume more system resources, such as memory and CPU processing. If we consider that a Web Server is indeed used just for hosting a website, then we realize that there will be no problem if we use more of our server's resources to provide the service our clients expects to have. We also have to keep in mind that if we move our server into a different network we might have issues because of our customized configuration. I would recommend to test different combinations even if we think that it is somewhat exaggerated. Being paranoid is acceptable when it comes to security.

3.1.2 Apache Modules

Modules will enhance the functionality of Apache. There is almost every type of module for whatever extra function we would like to incorporate to our Web Server. Here we would implement and configure three different modules for different purposes. One of them, mod_rewrite, comes preinstalled with Apache, the other two, mod_evasive and mod_security, can be found on the web.

3.1.3 PHP

The PHP script language will allow us to provide dynamic content to our clients. When it is first installed, PHP is configured to consume quite some resources from our server adding unnecessary extra load to it. Therefore, the PHP configuration file will be configured and optimized to reduce system resources consumption without sacrificing its functionality. A PHP script accelerator will also be implemented to help with this.

3.1.4 MRTG

The Multi Router Traffic Grapher will be responsible for monitoring our Web Server inbound and outbound traffic whether it is legitimate or malicious. There is no need for a special configuration.

3.1.5 APF

A firewall that will help to drop malicious or malformed packets as well as blocking IP addresses that exceeds the number of requests per second. This will improve our Web Server security and legitimate packet handling. In order to be efficient, we will have to be analyzing the access log from our server and add a custom rule to our iptables.

3.1.6 SSH

In a distributed defense, we need to be able to communicate with our defense nodes in real time and using the least resources possible from our server. Secure Shell is a program to securely log in into another computer and, in this case, to copy our logs from our Web Server to a defense node so they can see where the attacks are coming from.

3.1.7 ZmbScap

Zombie Scapper is an automated tool that detects and stops DDoS programs. This tool will be used by a defense node in a case where a distributed defense is required.

3.2 Optional Components

There are more modules and software available that could be used for this experiment, but they will not be tested due to system and network limitations. Despite of this, a list of suggested software is provided next for those who want to implement them. Again, all of this software can be found on the Internet and is open source.

- Mod_throttle
- Mod_balance

- Mod_proxy_balancer
- PHPids
- Monowall
- SNORT

4. Detecting and Defending Against DoS and DDoS Attacks

4.1 Detecting Methods

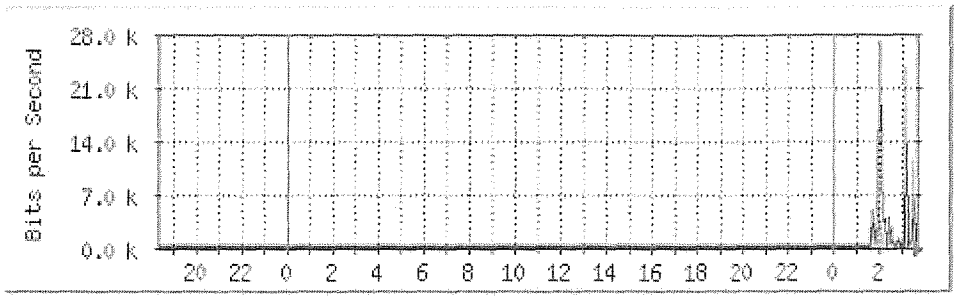
Before attempting to stop a DoS or DDoS attack we have to be sure which kind of attack we are receiving. With MRTG, Netstat, and Apache Logs it is quite easy to identify what the attacker is doing to flood the server. To quickly analyze these tools is vital when the server is under a DoS attack. Next we will see how these tools helped to create a proper defense in our SOHO experimental network.

4.1.1 MRTG

Using the traffic grapher as the primary tool for detecting DoS/DDoS attacks will give us an idea to decide whether we have to continue using other tools for detecting a possible attack or not.

Normal traffic should never exceed of our incoming limitations, unless our website has big popularity or when being affected by a DoS/DDoS attack.

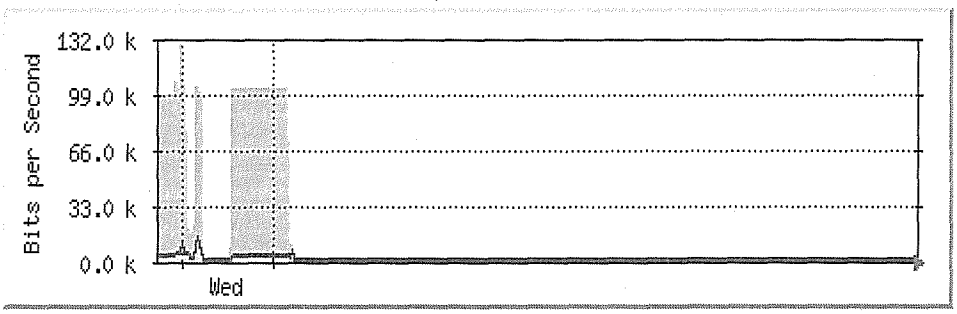
If the site traffic report shows excessive incoming connections everyday at a certain time then it is probably that a botNET is being used to attack the site.



	Max	Average	Current
In	23.6 kb/s (0.4%)	592.0 b/s (0.0%)	1848.0 b/s (0.0%)
Out	26.7 kb/s (0.4%)	328.0 b/s (0.0%)	424.0 b/s (0.0%)

Figure 5. Checking Traffic with MRTG – Normal State

When the inbound traffic is excessive then this could result in a DoS. In Figure 6 the website is receiving a tremendous incoming connections per second and this is when is a good idea to go and check our logs and netstat open connections in port 80 to determine wheter it is a DoS attack or not.



	Max	Average	Current
In	128.4 kb/s (2.0%)	12.7 kb/s (0.2%)	432.0 b/s (0.0%)
Out	14.4 kb/s (0.2%)	672.0 b/s (0.0%)	128.0 b/s (0.0%)

Figure 6. Checking Traffic with MRTG – Excessive Incoming Connections

4.1.2 Netstat

A simple command-line tool that displays established connections in our server in real time. The only problem here is when trying to get the connections because it has to be done in the very precise moment in which the user is requesting an item from our website, otherwise we would just have a "blank" output like the next figure:

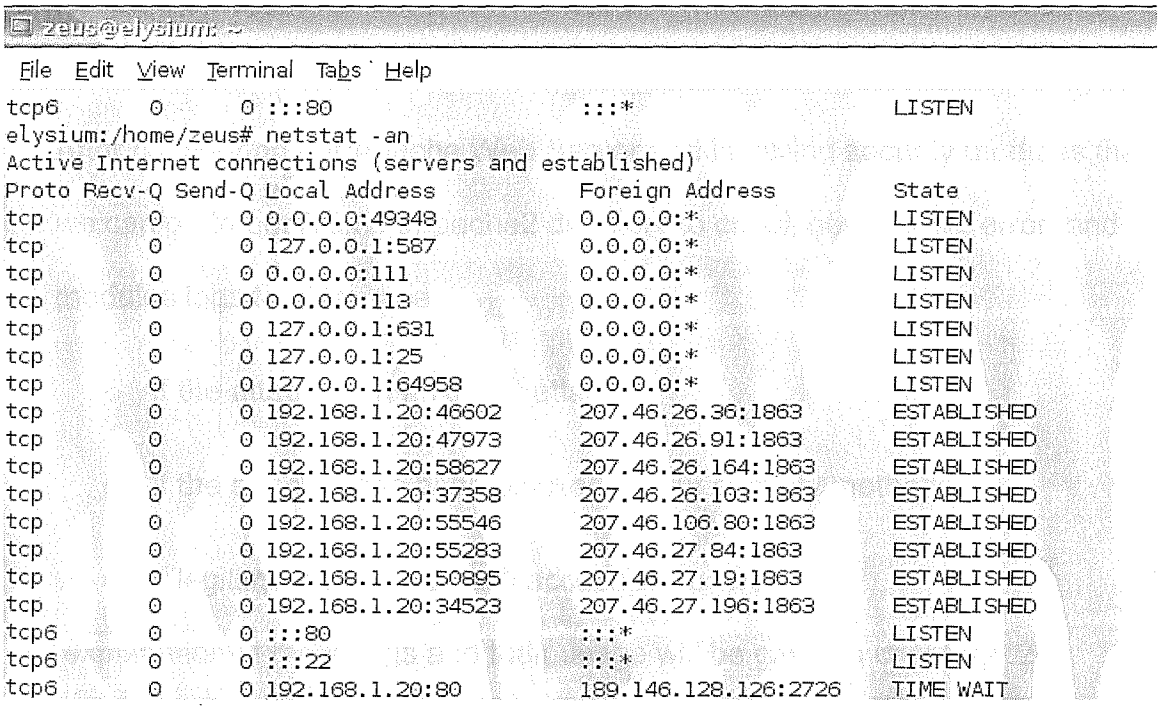


Figure 7. Network Statistics in a Normal Web Server

But when we are receiving a DoS or DDoS attack then the netstat will show all the established connections with the attacker at the first run. This is because an attacker does not request info from the server just once but several times per second. The status is often "TIME WAIT" as shown in Figure 8.

tcp	0	0 127.0.0.1:64958	0.0.0.0:*	LISTEN
tcp	0	0 192.168.1.20:46602	207.46.26.36:1863	ESTABLISHED
tcp	0	0 192.168.1.20:47973	207.46.26.91:1863	ESTABLISHED
tcp	0	0 192.168.1.20:58627	207.46.26.164:1863	ESTABLISHED
tcp	0	0 192.168.1.20:37358	207.46.26.103:1863	ESTABLISHED
tcp	0	0 192.168.1.20:55546	207.46.106.80:1863	ESTABLISHED
tcp	0	0 192.168.1.20:55283	207.46.27.84:1863	ESTABLISHED
tcp	0	0 192.168.1.20:50895	207.46.27.19:1863	ESTABLISHED
tcp	0	0 192.168.1.20:34523	207.46.27.196:1863	ESTABLISHED
tcp6	0	0 :::80	:::*	LISTEN
tcp6	0	0 :::22	:::*	LISTEN
tcp6	0	0 192.168.1.20:80	189.146.128.126:2726	TIME_WAIT

Figure 8. Detecting Attacker's IP Address with Netstat

4.1.3 Apache Logs

After configuring our Apache Web Server and installing security modules then we can go to our /var/log/apache2 directory to check our access, error, and modules logs to determine:

- If the attack is an IFRAME attack
- If the attacker has been blocked by our security methods
- If legitimate users can still access to our website

An explanation of these logs and their usage will be given in chapter 4.2.3.

4.2 Defending Methods

After gathering all the required information from the traffic analysis it is now possible to customize our Web Server and defense methods to try stopping the DoS/DDoS attack.

4.2.1 Optimizing and Configuring Apache

Since the Web Server has a default limit to handle incoming requests it is necessary to modify this settings so the server can do more requests in less time. The settings to be modified in the server, not in the modules yet, are:

- Time limit before killing connections: No need to wait so long and waste resources in not-responding connections.
- Maximum Server Child Instances: In order to be able to attend more clients then we have to open more branches to help deal with this problem.
- Maximum Client Handling: The default number for maximum requests that the server can process is not enough when dealing with a popular website or a DoS/DDoS attack, therefore this client handling limit has to be increased.
- Hide Apache information for security.

It is a good idea to try different settings with different types of tests before deciding a permanent configuration, although these settings can be modified at any time without difficulties.

In the computer used for this document, the ideal configuration ended like this:

```
Timeout 35
#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On
#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 100
#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 15
##
## Server-Pool Size Regulation (MPM specific)
##
# prefork MPM
# StartServers: number of server processes to start
# MinSpareServers: minimum number of server processes which are kept spare
# MaxSpareServers: maximum number of server processes which are kept spare
# MaxClients: maximum number of server processes allowed to start
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule mpm_prefork_module>
    ServerLimit          500
    StartServers          16
    MinSpareServers       13
    MaxSpareServers       28
    MaxClients           350
    MaxRequestsPerChild   0

# ============  Thesis CONF ==============

# Limits the version information the server gives.
ServerTokens Prod
ServerSignature Off

ServerName elysium
DirectoryIndex home.php
#blocks access to specific directories
<Directory "/var/www/dbms/mrtg">
  order allow,deny
allow from 127.0.1.1
</Directory>

<Directory "/var/www/dbms/webalizer">

<Directory "/var/www/dbms/_notes">

<Directory "/var/www/dbms/_mmServerScripts">

<Directory "/var/www/dbms/Connections">
```

Figure 9. apache2.conf Optimized

We could have increased these numbers but due system specifications (Appendix C) the server is not able handle a higher configuration. Higher configurations are for High End Server Systems [17].

4.2.2 Optimizing PHP

The settings in PHP needs to be configured so it consumes less system resources, prevents system scripts and files from being executed remotely, and uses a script accelerator for better performance. Function restrictions may be the only thing that it has to be checked if the configuration file is going to be copied in different servers.

```
;PHP modified conf. - Thesis

expose_php = Off
;don't show php version

; Resource Limits ;
max_execution_time = 30 ;maximum time for executing a script
memory_limit = 8M  ; script memory limit

;some PHP functions could permit the execution of a system command
;which is a security issue that it has to be fixed

disable_functions =exec,system,shell_exec,readfile
```

Figure 10. php.ini Optimized

4.2.3 Apache Modules

4.2.3.1 Mod_rewrite

Mod_rewrite is used to rewrite an URL requested by the client. IFRAMES can be used to amplify a DDoS attack, or they can be used to install malware into a computer, such a bot agent, making this computer part of a botNET [18].

When loaded and properly configured, mod_rewrite will stop IFRAME attacks. To create a rule for rewriting an URL in an IFRAME attack it is necessary to check the access log to locate the referrer and then we can load a custom rule to stop the attack.

Another useful implementation of this module is to protect the server bandwidth by preventing other sites from direct-linking to the files hosted by our server. Creation of rules that prevents these direct-links can be achieved by following the Rewrite rules [19].

4.2.3.2 Mod_evasive

Depending on the server sensitiveness of protection this module can be modified at anytime if desired. Again, in a SOHO network, like the one used in this document, the module has to be very cautious when it comes to detect DoS attacks.

Lowering the restrictions to make our security more sensitive against DoS attacks is only effective with Web Servers that does not often receives too much traffic. If the security is very sensitive on a website that has lots of people visiting everyday then they all might get banned from the server because the module will incorrectly think they are all trying to do a DoS attack.

Trying different configurations before setting up a permanent level of security is the best way to ensure effectiveness in the server security system.

```
<IfModule    mod_evasive20.c>
DOSHashTableSize 3097
DOSPageCount 1
DOSSiteCount 50
DOSPageInterval 1
DOSSiteInterval 1
DOSBlockingPeriod 15000
#ips de googlebot
DOSWhitelist 66.249.65.*
DOSWhitelist 66.249.66.*
DOSWhitelist 66.249.67.*
DOSWhitelist 66.249.70.*
DOSWhitelist 66.249.72.*
DOSWhitelist 66.249.73.*
DOSWhitelist 66.249.85.*
DOSLogDir         /var/log/apache2/evasive.log
</IfModule>
```

Figure 11. Configuration of mod_evasive.

After someone attempts to do a DoS attack by requesting multiple requests per second then our module will automatically blacklist the IP address for a predetermined period of time. All of the possible attacks are saved in the module log for further reference.

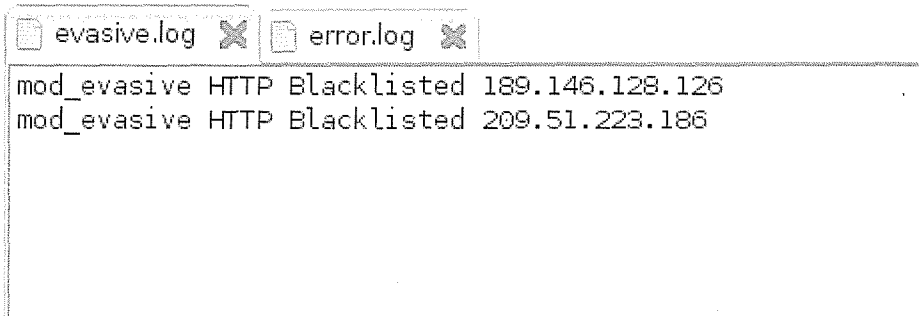


Figure 12. Example of Blacklisted IP Addresses Logged by mod_evasive.

After we have discovered our attackers IP address with previous methods then we can configure mod_evasive to deny access to our web application.

Here is an example of two attackers who got denied after trying to flood the server with multiple requests:

```
[Thu Mar 27 01:32:17 2008] [error] [client 189.146.128.126] File does not exist: /var/www/dbms/favicon.ico
[Thu Mar 27 01:34:19 2008] [error] [client 209.51.223.186] File does not exist: /var/www/dbms/45
[Thu Mar 27 01:34:21 2008] [error] [client 209.51.223.186] File does not exist: /var/www/dbms/favicon.ico, referer: /
[Thu Mar 27 01:34:46 2008] [error] [client 209.51.223.186] File does not exist: /var/www/dbms/favicon.ico, referer: /
[Thu Mar 27 01:34:48 2008] [error] [client 209.51.223.186] File does not exist: /var/www/dbms/45
[Thu Mar 27 01:34:49 2008] [error] [client 209.51.223.186] File does not exist: /var/www/dbms/favicon.ico, referer: /
[Thu Mar 27 01:35:08 2008] [error] [client 189.146.128.126] File does not exist: /var/www/dbms/4555
[Thu Mar 27 01:35:11 2008] [error] [client 189.146.128.126] File does not exist: /var/www/dbms/4555
[Thu Mar 27 01:35:11 2008] [error] [client 189.146.128.126] client denied by server configuration: /var/www/dbms/4555
[Thu Mar 27 01:35:12 2008] [error] [client 189.146.128.126] File does not exist: /var/www/dbms/4555
[Thu Mar 27 01:35:12 2008] [error] [client 189.146.128.126] client denied by server configuration: /var/www/dbms/4555
[Thu Mar 27 01:35:13 2008] [error] [client 189.146.128.126] client denied by server configuration: /var/www/dbms/4555
[Thu Mar 27 01:35:13 2008] [error] [client 189.146.128.126] client denied by server configuration: /var/www/dbms/4555
[Thu Mar 27 01:35:14 2008] [error] [client 189.146.128.126] client denied by server configuration: /var/www/dbms/4555
[Thu Mar 27 01:35:15 2008] [error] [client 189.146.128.126] client denied by server configuration: /var/www/dbms/4555
[Thu Mar 27 01:35:15 2008] [error] [client 189.146.128.126] client denied by server configuration: /var/www/dbms/4555
[Thu Mar 27 01:35:17 2008] [error] [client 189.146.128.126] client denied by server configuration: /var/www/dbms/
[Thu Mar 27 01:35:18 2008] [error] [client 189.146.128.126] client denied by server configuration: /var/www/dbms/
[Thu Mar 27 01:35:19 2008] [error] [client 189.146.128.126] client denied by server configuration: /var/www/dbms/
[Thu Mar 27 01:35:19 2008] [error] [client 189.146.128.126] client denied by server configuration: /var/www/dbms/
[Thu Mar 27 01:37:36 2008] [error] [client 209.51.223.186] File does not exist: /var/www/dbms/favicon.ico, referer: /
[Thu Mar 27 01:38:18 2008] [error] [client 209.51.223.186] File does not exist: /var/www/dbms/favicon.ico, referer: /
[Thu Mar 27 01:41:26 2008] [error] [client 189.146.128.126] client denied by server configuration: /var/www/dbms/
[Thu Mar 27 01:41:37 2008] [error] [client 209.51.223.186] client denied by server configuration: /var/www/dbms/
[Thu Mar 27 01:41:38 2008] [error] [client 209.51.223.186] client denied by server configuration: /var/www/dbms/
[Thu Mar 27 01:42:16 2008] [error] [client 209.51.223.186] client denied by server configuration: /var/www/dbms/
[Thu Mar 27 01:43:34 2008] [error] [client 209.51.223.186] client denied by server configuration: /var/www/dbms/
[Thu Mar 27 01:49:43 2008] [error] [client 189.146.128.126] client denied by server configuration: /var/www/dbms/
[Thu Mar 27 01:50:15 2008] [error] [client 189.146.128.126] client denied by server configuration: /var/www/dbms/
[Thu Mar 27 01:50:16 2008] [error] [client 189.146.128.126] client denied by server configuration: /var/www/dbms/
[Thu Mar 27 01:50:17 2008] [error] [client 189.146.128.126] client denied by server configuration: /var/www/dbms/
[Thu Mar 27 01:50:31 2008] [error] [client 189.146.128.126] File does not exist: /var/www/dbms/damepermisohahhaha
[Thu Mar 27 01:50:33 2008] [error] [client 189.146.128.126] client denied by server configuration: /var/www/dbms/
[Thu Mar 27 01:52:41 2008] [error] [client 189.146.128.126] client denied by server configuration: /var/www/dbms/
```

Figure 13. IP Addresses Blocked After Attempting Multiple Requests.

Notice that the attacker is not trying to get any valid data from the server, his only purpose is to disrupt the server communication by flooding its bandwidth.

4.2.3.3 Mod_security

The configuration for this module is very dynamic. There are no default values that we can use, however, if the module is properly configured then the server can block IP addresses that attempt to breach our security. For example, the attacker may want to try accessing to the server's system using a Remote File Inclusion (RFI) method.

Remote File Inclusion attacks allow malicious users to run their own PHP code on a vulnerable website. The attacker is allowed to include his own malicious code in the space provided for PHP programs on a web page. This allows the attacker to include any remote file of his choice simply by editing the URL. Attackers commonly include a malicious PHP script called a webshell, also known as a PHP shell. A webshell can display the files and folders on the server and can edit, add or delete files, among other tasks. Scripts that send Spam are also very common. Potentially, the attacker could even use the webshell to gain administrator-level, or root, access on the server [20].

Mod_security can prevent this type of attack by analyzing the requested URL and denying access to the attackers IP address for a predetermined period of time. Figure 14 shows an example of a possible RFI and how the attacker is forbidden by mod_security.

```
Request: 24.109.91.45 189.146.128.126 - - [20/Mar/2008:22:49:11 +0500] "GET /index.php?action=http://hotraebywka.chat.ru/images/girl? HTTP/1.1" 403 278
"-" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322)" - "-"
Handler: application/x-httpd-php
---------------------------------------
GET /index.php?action=http://hotraebywka.chat.ru/images/girl? HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322)
Accept: text/html, */*
Content-Type: text/html
Host: elysium
Connection: keep-alive
mod_security-action: 403
mod_security-message: Access denied with code 403. Pattern match "\\.php\\?(((LOCAL|INCLUDE|PEAR|SQUIZLIB)_PATH|action|content|dir|name|menu|pn_path|
pagina|path|pathtoroot|cat|include_location|gorumDir|root|page|site|topside|pun_root|open|seite)=(http|https|ftp)\\:/|.*(cmd|command)=(cd|\\;|perl |python
|rpm |yum |apt-get |emerge |lynx |links |mkdir |elinks |cmd|pwd|wget |lwp-(download|request|mirror|rget) |id|uname |cvs |svn |(s|r)(cp|sh) |net(stat|cat)|
rexec |smbclient |t?ftp |ncftp |curl |telnet |gcc |cc |g\\+\\+ |whoami|\\./|killall |rm \\-[a-z|A-Z]))" at REQUEST_URI [severity "EMERGENCY"]

HTTP/1.1 403 Forbidden
Content-Length: 278
Keep-Alive: timeout=3, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=iso-8859-1
--5863aa7d--

--21d40e29-----------------------------
Request: 24.109.91.45 209.51.223.186 - - [20/Mar/2008:22:52:36 +0500] "GET /index.php?action=http://minaru.hut2.ru/images/cs.txt? HTTP/1.1" 403 278 "-"
"Wget/1.1 (compatible; i486; Linux; RedHat7.3)" - "-"
Handler: application/x-httpd-php
---------------------------------------
GET /index.php?action=http://minaru.hut2.ru/images/cs.txt? HTTP/1.1
User-Agent: Wget/1.1 (compatible; i486; Linux; RedHat7.3)
Accept: text/html, */*
Content-Type: text/html
Host: elysium
```

Figure 14. Possible RFI detected by mod_security

4.2.4 APF

The firewall will give the server an extra protection against DDoS tools used by the attackers. APF uses a list of commonly IP addresses and ports involved in DDoS attacks. The IP addresses are blocked and the ports are closed in the server. The list is updated every time the firewall is executed. The server administrator can take advantage of Apache, mod_evasive, and MRTG to create custom rules in iptables so they can be used by APF against DoS/DDoS attacks [21].

4.3 Distributed Defense

A distributed defense requires two or more computers running a Linux/Unix OS. The nodes shares vital security information about each other computer connected in this defense network, therefore the extra nodes used in this defense must be from trustworthy persons. Some information shared among these nodes is:

- Root password.
- Apache version.
- Modules enabled.
- Ports opened/closed.
- Permission to remotely execute commands.

When a DDoS attack is affecting the Web Server then we have to try to make this fight even. In example, if three zombies are attacking our Web Server then the ideal would be asking other three or four computers to help stopping this attack. All the computers share system information so they will all know the attacker IP address.

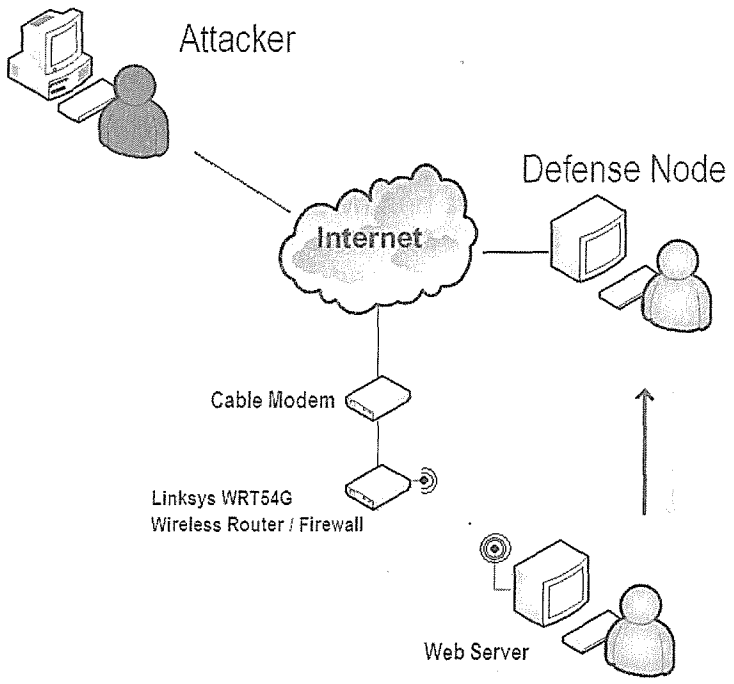


Figure 15. Distributed Defense System.

One method to inform other nodes about a possible attack is to configure mod_evasive and install a SMTP server so the module can send an email [22] to other defense nodes whenever an attempt of disruption occurs. The access logs can also be included.

Once the defense node receives the IP addresses of the attackers then a counter-attack can be started to try stopping the DDoS attack.

A tool that can be used by the defense node is ZmbScap [23]. Assuming that the attackers are zombies, it means these computers are vulnerable to other threats and therefore we can gain access to them and counter attack more effectively. The problem here is that is not a legal solution and we can not do this without consent of the zombie computer owner.

Analyzing traffic and viewing the log files in group is much better than doing it by one person. A "position" for each node should also be considered, in example, 2 nodes can be working trying to stop zombies while other 2 focus on tracing the real attacker to bring it down and stop the attack.

After the situation has been handled there are many things we have to take in consideration. The first thing to do is report the attacker to our ISP so they can take legal actions to prosecute the attacker.

Next is to see how much damage the server received. Checking log files and MRTG graphs will help the Web Server administrator to locate weak points in the server security and enforce them for future attacks.

Changing the IP address is something to have in consideration. The problem is that if the attacker wants to do another DDoS attack to us then he probably will use the same IP address he used the first time he attacked. This will result into an unusable IP address for whoever gets this address next time the ISP assigns it [24].

Also, having another backup Web Server ready in the case of a total denial of service is a solution.

5 Conclusions

The server used in this experiment is a common computer that can be found nowadays in home and offices. It is a computer that is not intended to be a Web Server but it is capable of that. Therefore the ability for handling DDoS attacks is reduced because of its specifications. For DoS attacks there were no problems at all.

Denial of Service attacks can be handled quite easy and we can continue offering our services even under attack. The attacker needs a greater bandwidth than the victim in order to be able to flood but since the confrontation is between two computers this issue can be handled by dropping packets and block requests from the attacker. If the Web Server has a higher bandwidth than the attacker then it will be impossible to produce a DoS attack but a DDoS attack can still be done.

In the case of a Distributed Denial of Service attack the server must be completely dedicated to host the website and not wasting any resources in unnecessary applications, in example, playing an online game. All the system resources are vital for the server to keep running during a DDoS attack. These system resources are not only network related. The server hardware is also an important factor to take in consideration for handling multiple requests. A High End Server Computer is likely to be more efficient than a Laptop as a Web Server. The LAN in which the Web Server is installed is another important aspect that affects or benefits the server

ability to handle incoming requests. For example, suppose our ISP is offering its service through fiber optic directly to our location, if we have the server connected through a wireless router instead of having it connected directly to the fiber optic then certainly the server is wasting bandwidth and therefore is not sending and receiving data as the way it is supposed to be meant transferred.

When doing the configuration for either Apache Web Server or Apache modules it is indispensable to keep in mind how much can the server process. To properly configure the Web Server requires time and lots of tests. Exceeding consumption of resources can result in unexpected behavior from the website. Using all the resources does not mean the server will do everything much faster or accurately, there always has to be some kind of system reserve.

Mastering the modules, traffic analyzers, and log events will greatly improve the chances to handle an attack whether it is a DoS or a DDoS.

Learn from the attacker. Not all the attacks are made in exactly the same way. Crackers will keep using different methods in order to achieve their goals, and as the Internet keeps growing so the attacks will. Every new attack will be a new experience. Know that your enemy has no rules, grab what they show you and use it against them.

The only real solution and prevention of such attacks is to instruct people about computer hygiene. Not everyone knows how to prevent catching a virus or a Trojan horse, or how to remove them once they have

gotten infected. Preventing people from accessing harmful websites is something that is actually being done without the need of any special software. An example could be Google search engine which is starting to display a warning message under a website that is considered harmful. If people can start taking this in consideration as if it was their own health then it is possible to start thinking in a world without Denial of Service attacks.

6 Future Directions

- Use of a High End Server. A real Web Server is mounted on a High End computer system built specially for this application. The ability for defending against DoS attacks could be greatly improved if a real server is used and properly configured. Remember to use wisely the system resources and to experiment with several configurations.
- Faster and better Internet connection. The use of more sophisticated hardware and network components will help the server to handle bandwidth more efficiently, in example, using fiber optic instead of DSL. If there is intention to use this defenses in a small and mid-sized companies do not expect to have the same luck as other people. For these companies it could be necessary to use more than two servers and to apply a Load Balance for better traffic handling.
- Implementing a firewall in the Internet. It might be possible to use a firewall on the Internet to prevent malicious traffic from getting to the server IP address. In other words, to block the attack before a connection with the server is established. The ISP would have to agree to be involved in this experiment.
- NVIDIA GPU for packet processing. Recent academic papers and articles [25] claims that using a NVIDIA GPU to solve the most complex computation-intensive challenges such as oil and gas exploration, financial risk management, product design, medial imaging, and scientific research. If

the GPU can be used to process the traffic packets in a Web Server and, if possible, in the "Internet firewall" then there is a possibility to say that DoS attacks will no longer be an issue. Traffic packets could be quickly analyzed by the GPU to determine if they are either corrupt or malicious packets. This could also mean that there will be no extra use for the server CPU.

- Global Load Balance. Since a DDoS attack can be done from any part of the world where Internet is available, a Global Load Balance might be an extra help for a distributed defense. Two issues arises from this:
    1. Direct access to our virtual server might be restricted and therefore defending methods could not be implemented.
    2. Is not a free service.

BIBLIOGRAPHY

[1] "Knoppix Hacks: Scan for Viruses" in http://whitepapers.techrepublic.com.com/abstract.aspx?docid=323343 Tech Republic, October 2004.

[2] "DoS and DDoS Attack Guide" in http://www.dos-attacks.com/ Faisal Khan, 2008.

[3] "CERT/CC Denial of Service" in http://www.cert.org/tech_tips/denial_of_service.html Carnegie Mellon University, 2001.

[4] "Denial of Service Attack (DoS)" in http://www.mountainwave.com/avcenter/venc/data/dos.attack.html Symantec Corportaion, NA.

[5] "Understanding Denial-of-Service Attacks" in http://www.us-cert.gov/cas/tips/ST04-015.html Mindi McDowell, August 2007.

[6] "Factsheet – Root Server Attack on 6 February 2007" in http://www.icann.org/announcements/factsheet-dns-attack-08mar07.pdf ICANN, March 2007

[7] "Cracker" in http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci211852,00.html Hans Sjoholm, June 2007.

[8] "Estonia and Russia – A cyber-riot" in http://www.economist.com/world/europe/displaystory.cfm?story_id=9163598 The Economist, May 10, 2007.

[9] "Information about DDoS and Denial of Service Attacks" in http://www.ddosinfo.com/ NA, NA.

[10] Molist, M. "Entrevista El-Brujo" Revista @rroba, vol. 116, pp. 38-42, 2007.

[11] "It Can Happen to You" in http://www.prolexic.com/news/20071102-security.php Prolexic Technologies, November 2, 2007.

[12] "IIJ Releases the IIJ DDoS Solution Service; Management of DDoS Detection and Protection Hardware Protects Corporate Networks from Large-Scale Attacks" in http://findarticles.com/p/articles/mi_m0EIN/is_2005_Oct_27/ai_n15755366 Business Wire. Oct 27, 2005.

[13] "RioRey :: The DDoS Specialist" in http://www.riorey.com/ RioRey, Inc., 2008.

[14] "Operating System Market Share for February, 2008" in http://marketshare.hitslink.com/report.aspx?qprid=8&qpmr=100&qpdt=1&qpct=3&qptimeframe=M&qpsp=109&qpnp=1 Net Applications.com, 2008.

[15] "Secunia: Stay Secure 2007 Report" in http://secunia.com/gfx/SECUNIA_2007_Report.pdf Secunia, 2007.

[16] "Linux Stability for Web Server" in http://www.melbourneit.com.au/cc/dedicated-servers/linux-servers Melbourne IT, 2007.

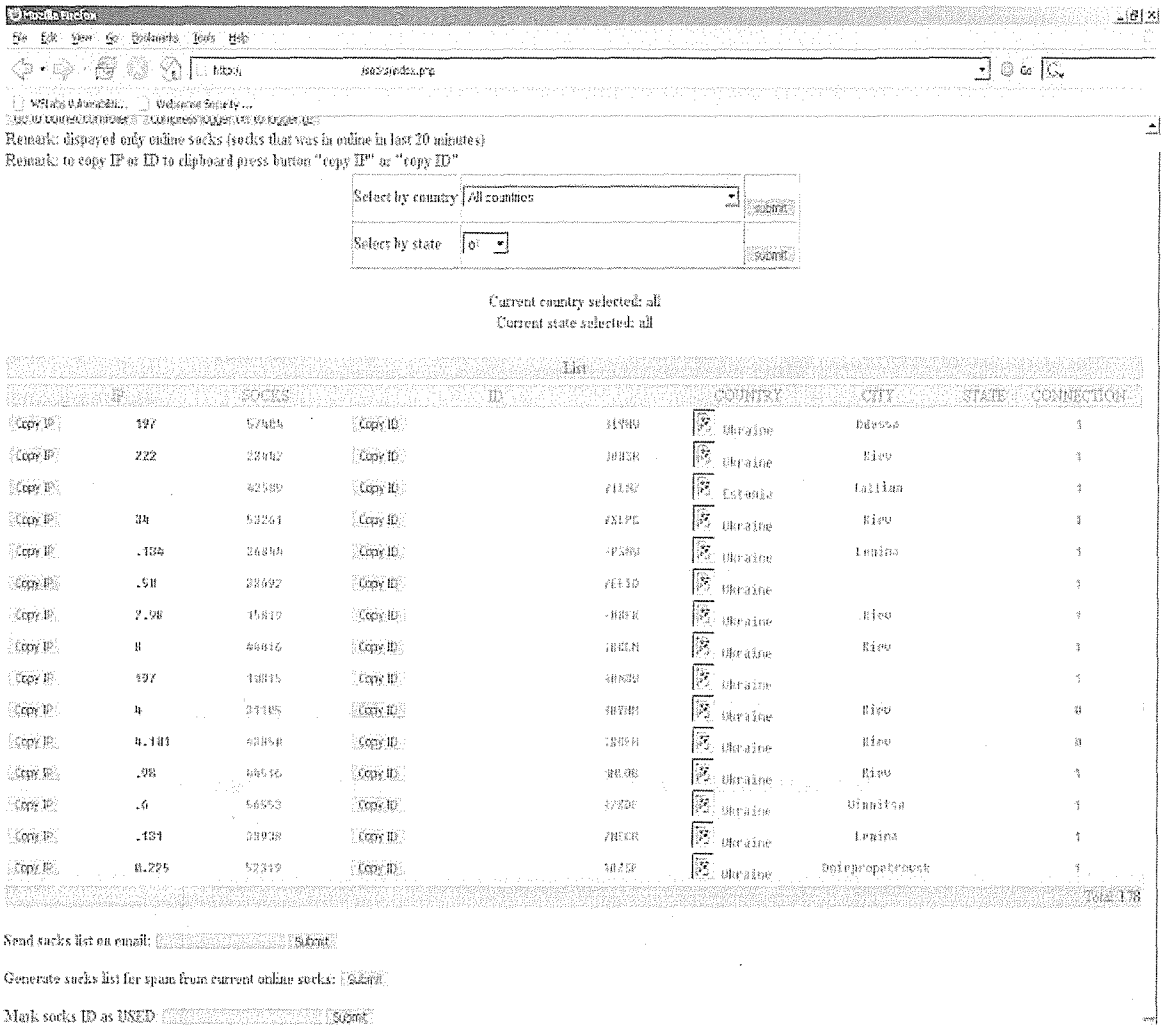
[17] "Apache Performance Notes" – Hardware and Operating System Issues" in http://httpd.apache.org/docs/2.2/misc/perf-tuning.html The Apache Software Foundation, 2008.
[18] Keizer, G. "Hackers Launch Massive IFRAME attack" in http://www.pcworld.idg.com.au/index.php/id;26001482 Computerworld , March 14, 2008
[19] "URL Rewriting Guide – Apache HTTP Server" in http://httpd.apache.org/docs/2.0/misc/rewriteguide.html Ralf S. Engelschall, December 1997.
[20] "Remote File Inclusion" in http://en.wikipedia.org/wiki/Remote_File_Inclusion, March 11, 2008.
[21] "R-fx Networks – Internet Security Solutions - APF" in http://rfxnetworks.com/apf.php R-fx Networks, NA.
[22] "Open Proxy Honey Pots – Mod_DOSevasive" in http://honeypots.sourceforge.net/open_proxy_honeypots.pdf Ryan C. Barnett, March 30, 2004.
[23] "ZmbScap – Zombie Scapper" in http://zmbscap.sourceforge.net/ Metaeye Security Group, NA.
[24] "Security Now! – Denial of Service Attacks" in http://media.grc.com/sn/SN-008-lq.mp3 Security Now! Ep. 8, Steve Gibson, 2005.
[25] "CUDA Showcase – NVIDIA" in http://www.nvidia.com/object/cuda_showcase.html NVIDIA Corporation, 2008.

Appendix A. Web based botNET

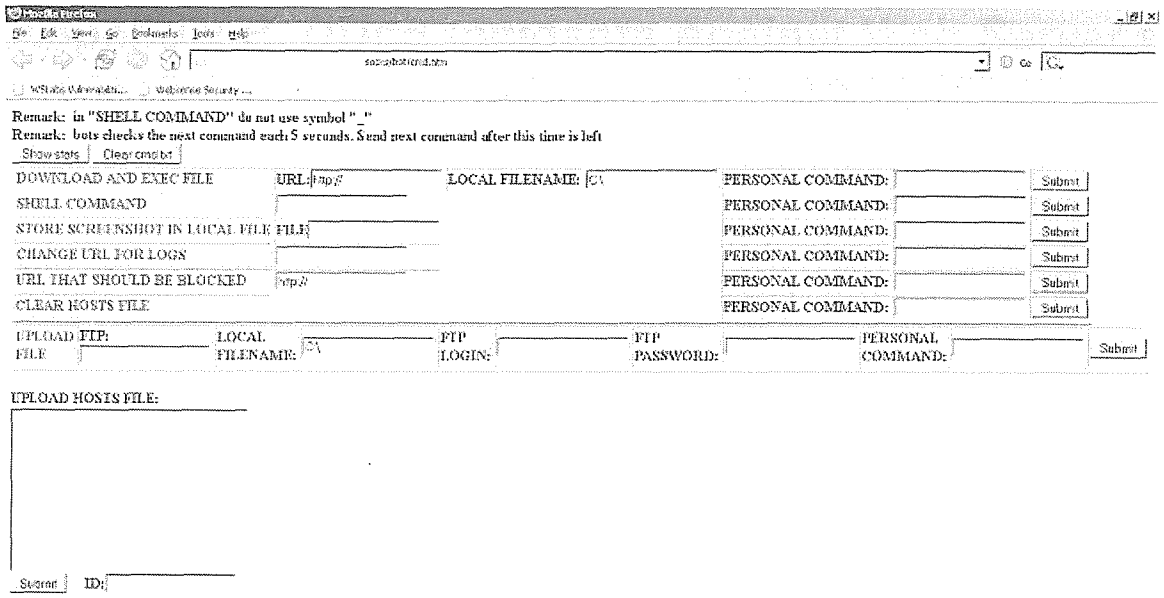
Websense® Security Labs™ continues to research additional tactics that botnet operators utlize in order to command and control (C&C) their infected Zombies on the internet. This research has found more frequent use of web-based controllers. The currently most used method to control a botnet is through IRC, where commands can be sent to the infected hosts. The information is transmitted to and from the hosts.

The addition of using HTTP to control the bots and trigger them to upload their information creates another area where security professionals need to investigate possible infections and takedowns. Websense Security Labs has seen this tactic as particularly popular with bots that are used to capture and transmit keylogging programs information and to store account information.

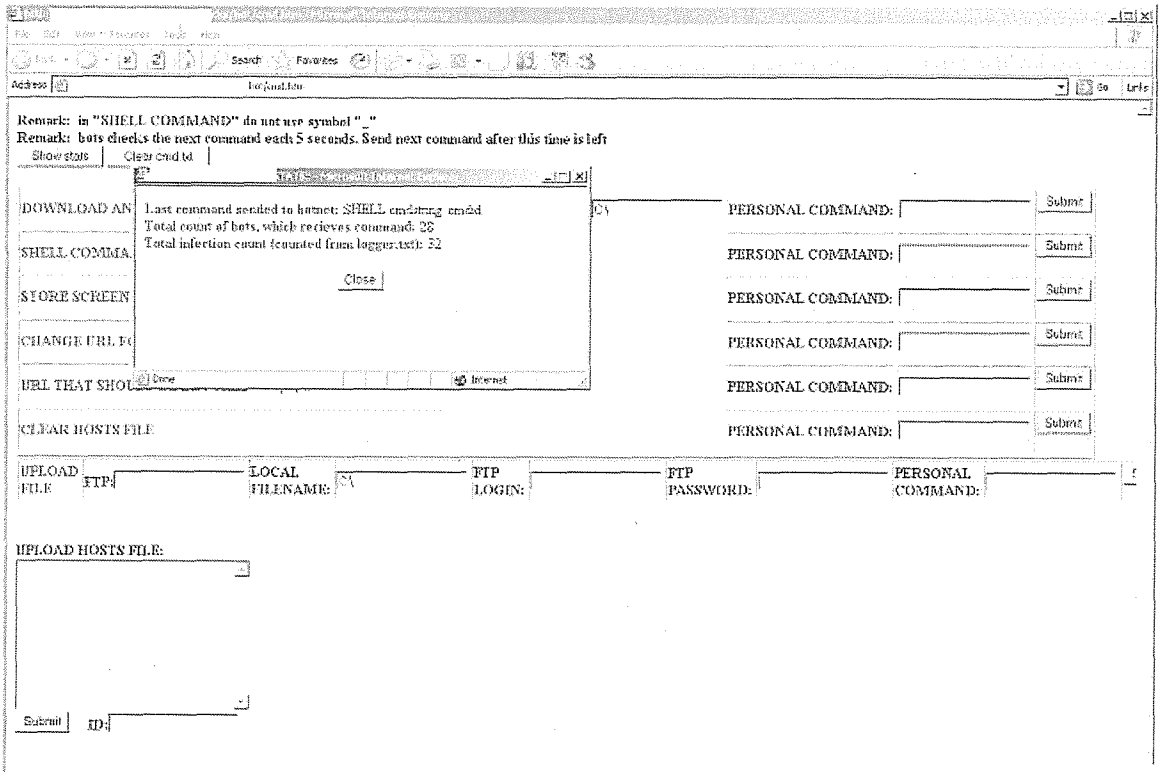
The screenshots below show an example of often used web-based botNET controller. The first screen gives the operator the ability to view all the infected hosts and display them by country and city, IP address, and unique ID.



The second screenshot is an example of how the controller can send commands to the infected hosts and modify the machine. In this screen, you can see how the operator can block URLs that they do not want the machine to contact, such as anti-virus update centers and Microsoft updates. They can modify the "hosts" file to redirect traffic, such as modify well-known banking and ecommerce sites to be redirected to fraudulent site. By a mouse click, they can send programs and commands to launch on the remote machines.



The third screenshot shows statistical information on how many infected hosts have received the input information, i.e. the success rate.



Appendix B. DDoS Incidents Against Web Servers

"A system may also be compromised with a trojan, allowing the attacker to download a zombie agent (or the trojan may contain one). Attackers can also break into systems using automated tools that exploit flaws in programs that listen for connections from remote hosts. This scenario primarily concerns systems acting as servers on the web." – Denial-of-Service Attack – Distributed Attack in http://en.wikipedia.org/wiki/Denial-of-service_attack

"CastleCops has been the target of more than a dozen DDoS attacks" - It Can Happen to You in http://www.prolexic.com/news/20071102-security.php

"The WordPress.com blog-hosting service suffered a denial-of-service (DoS) attack that began Saturday and was still preventing users from logging in or posting to their blogs on Tuesday." -
DoS Attack Prevents Access to WordPress.com blogs in
http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9063440

""Toshiba Australia has confirmed it suffered a denial of service attack on its Web" - 22 February 2001, Toshiba Latest DoS Victim in
http://www.zdnet.com.au/news/security/soa/Toshiba-latest-DoS-victim/0,130061744,1202 05189,00.htm

"Many corporate websites have suffered from illegal denial-of-service (DoS) attacks more than once. The companies that learn how to turn these experiences to their advantage go a long way to ensuring it doesn't happen again." – Best Practices for Preventing Denial of Service (DoS) Attacks in http://www.microsoft.com/technet/archive/security/bestprac/dosatack.mspx

"With the growth of Internet and mobile connectivity, online gaming and mobile gaming are thriving. Due to their low value, high volume transactions, online sports sites are easy targets for criminal DDoS attacks." - Ensuring Availability of Internet Gaming and Mobile Gaming Sites in http://www.intruguard.com/documents/GamingWhitePaper_000.pdf

"The web site of anti-spyware activist Ben Edelman is back online after an extended outage, apparently caused by a distributed denial of service (DDoS) attack." - Spyware Activist Web Site Targeted By DDoS Attack in http://news.netcraft.com/archives/2005/02/09/spyware_activist_web_site_targeted_by_ddos_attack.html

Appendix C. Experimental System Specifications

Specifications

General
Computer Type: Notebook
Type of Use: Portable
Action Buttons: S1, S2 (programmable), Volume and Mute
Keyboard: QWERTY, 86 keys with 2.5mm stroke and 19.05mm pitch
Pointing Device: Electro-Static touch pad

Hardware
Docking Station: Docking station connector
Camera: Built-in camera and microphone

Processor
Type: Intel® Core™ 2 Duo Processor T5600
Speed: 1.83GHz
Front Side Bus Speed: 667MHz
L2 Cache: 2MB
Technology: Intel® Centrino® Duo Mobile Technology

Memory
Type: DDR2
Installed: 1.5GB (1024MBx1 - 512MBx1) PC2-4200
Maximum: 2GB
Speed: 533MHz

Hard Drive (External)
Capacity: 100GB
Speed: 5400rpm
Interface: IDE

Optical Drive #1
DVD+R DL Write: 4x max.
DVD+R Write: 8x max.
DVD+RW Write: 8x max.
DVD-R Write: 8x max.
DVD-RW Write: 6x max.
CD-R Write: 24x max.
CD-RW Write: 16x max.
DVD±RW: Yes
CD Read: 24x max.
DVD Read: 8x max.

DVD-R DL Write: 4x max.
DVD-RAM Read: 5x max.

Expansion Slots
Multimedia Card Reader: Memory Stick Duo™ media with MagicGate® functionality
ExpressCard™ /34 Slot 5-in-1 Memory Card Adaptor (VGP-MCA20) supporting Memory Stick®, Memory Stick PRO™ , Secure Digital, xD-Picture Card, and MultiMediaCard (MMC)
PCMCIA - Type II/Type I card slot with CardBus support

Audio
Sound System: Sony® Sound Reality™ - Audio Enhancer

Display
Screen or Display Technology: WXGA LCD
Screen Size: 15.4"
Resolution: 1280 x 800
XBRITE-ECO™ Technology

Graphics
Processor: Intel® Graphics Media Accelerator 950
Video RAM: 128MB [Dynamically Allocated Shared (RAM/Video) Memory]
Chipset: Intel® 945GM
Interface: VGA Out with Smart Display Sensor

Inputs and Outputs
Headphone Jack
i.LINK® Interface x1
Memory Stick® Media Slot
Microphone Input
S-Video Output(s) x1
USB Port x3 (2.0 compliant)
VGA Output

Networking/Modem
Ethernet Protocol: Fast Ethernet (RJ-45)
Ethernet Speed: 10Base-T/100Base-TX
Modem Type: Integrated V.92/V.90 Modem (RJ-11)
Wireless LAN: Intel® PRO/Wireless 3945ABG Network Connection (802.11a/b/g)

Power
Battery Type: Standard Lithium-ion Battery
Estimated Battery Life: 2.5-4 hours
Power Requirements: 110W+10%