Distributed Denial of Service Attacks: Explain nation, classification and suggested Solutions

Written By:

Ahmed Saafan

March 23, 2009

1. DDoS Attacks

“This page cannot be displayed”. Imagine that you keep on seeing this message while you are trying to access your e-banking, e-mail or even social network account with no hope. Imagine that this state of “denial” is not just for seconds or minutes, it lasted hours or days! This is a typical Distributed Denial of Service (**DDoS**) Attack.

1.1. Definition

A DDoS attack is defined as an attack in which a multitude of compromised systems attack a single target, thereby causing denial of service for users of the targeted system. The flood of incoming messages to the target system essentially forces it to shut down, thereby denying service to the system to legitimate users (Information Security Magazine, 2006).

To be able to conduct a successful DDoS attack, hackers use what is called a “**botnet**”. A botnet is a jargon term for a collection of software robots, or bots, that run autonomously and automatically. The term is often associated with malicious software but it can also refer to the network of computers using distributed computing software (Wikipedia, 2008).

A botnet is controlled by an originator or a “**botmaster**”. The botmaster utilizes autonomous malicious spreading software (**agents**) to acquire new victims (**zombies**). These zombies are the core of any DDoS attack. Zombies are usually innocent people’s computers; people who are unaware that they are part of an evil botnet. With agents running on the zombie machines, the botmaster just initiates an attack command and selects the target. The command is passed to the zombies and the zombies execute. The unaware target finds itself bombarded with tons of packets flooding its network and causing its denial of service. Table 1 shows top ten known botnets and their estimated size.

Name	**Est. Bot #**	**Spam Capacity**	**Aliases**
Conficker	9,000,000	10 billion/day	DownUp, DownAndUp, DownAdUp, Kido
Kraken	495,000	9 billion/day	Kracken
Srizbi	450,000	60 billion/day	Cbeplay, Exchanger
Bobax	185,000	9 billion/day	Bobic, Oderoor, Cotmonger, Hacktool.Spammer, Kraken
Rustock	150,000	30 billion/day	RKRustok, Costrat
Cutwail	125,000	16 billion/day	Pandex, Mutant (related to: Wigon, Pushdo)
Storm	85,000	3 billion/day	Nuwar, Peacomm, Zhelatin
Grum	50,000	2 billion/day	Tedroo
Onewordsub	40,000	NA	-
Mega-D	35,000	10 billion/day	Ozdok
Nucrypt	20,000	5 billion/day	Loosky, Locksky
Wopla	20,000	600 million/day	Pokier, Slogger
Spamthru	12,000	350 million/day	Spam-DComServ, Covesmer, Xmiler

Table 1 (Wikipedia, 2008)

Criminals are keen to recruit new machines to a botnet to create a resource that they can use or which can be hired out to other gangs.

Most spam or junk mail is routed through the hijacked machines forming a botnet. Currently, the vast majority of machines in these botnets are PCs running a version of Microsoft Windows. In June 2008, Shadow server Foundation knew about more than 100,000 machines that were part of a botnet. By the end of August this figure had exceeded 450,000 machines (BBC News, 2008). These numbers are not very accurate but they are "at least" numbers which give us an idea about the magnitude of the problem.

1.2. Types of DDoS Attacks

There are three main classifications for DDoS attacks. One based on the methodology of communication between the attacker and the victim. Another classification based on the spreading technique. And the last according to the exploitation mechanism itself. These classifications might not seem very obvious. However, they are a great way to understand all aspects of the botnets from different prespectives.

1.2.1. According to methodology of communication

There are two types of agents according to the methodology of communication with the victim; handler based agents and IRC based agents. Figure 1 summarizes the architecture of this type of networks very well.

1.2.1.1. Hander Based Agents

An Agent-Handler DDoS attack network consists of clients, handlers, and agents (see Figure 2). The client platform is where the attacker communicates with the rest of the DDoS attack network. The handlers are software packages located on computing systems throughout the Internet that the attacker uses to communicate indirectly with the agents. The agent software exists on compromised systems that will eventually carry out the attack on the victim system. The attacker communicates with any number of handlers to identify which agents are up and running, when to schedule attacks, or when to upgrade agents. Depending on how the attacker configures the DDoS attack network, agents can be instructed to communicate with a single handler or multiple handlers. Usually, attackers will try and place the handler software on a compromised router or network server that handles large volumes of traffic. This makes it harder to identify messages between the client and handler and between the handler and agents.

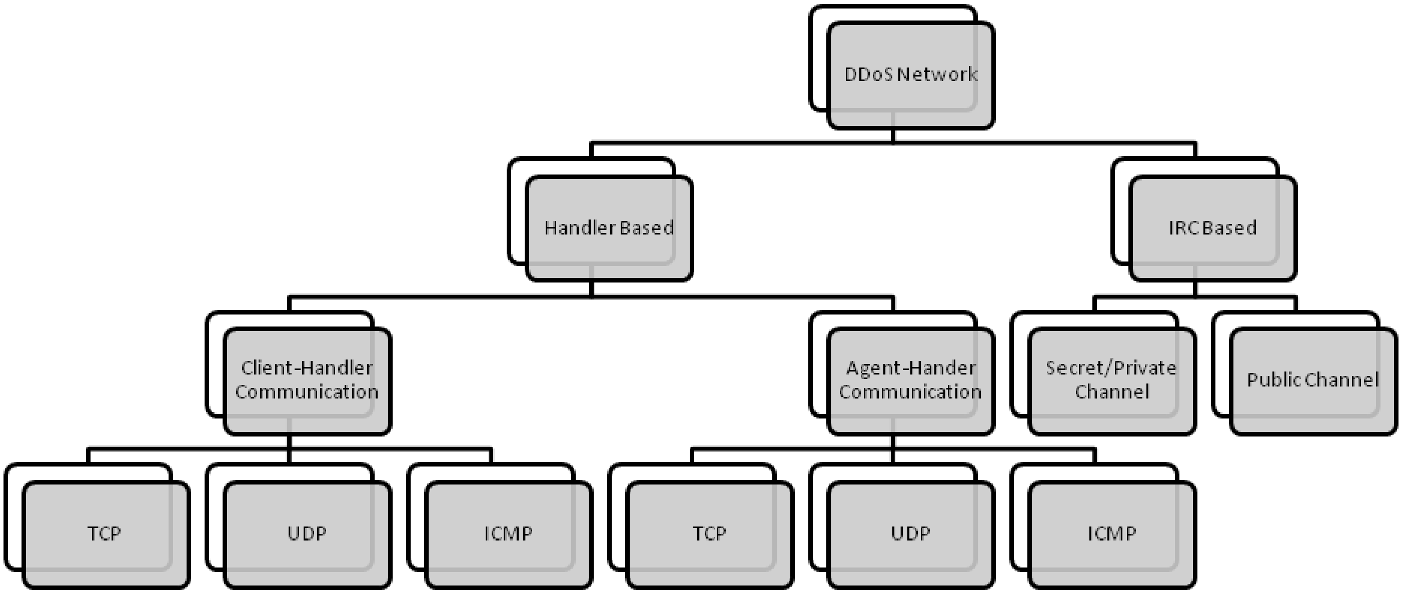


Figure 1

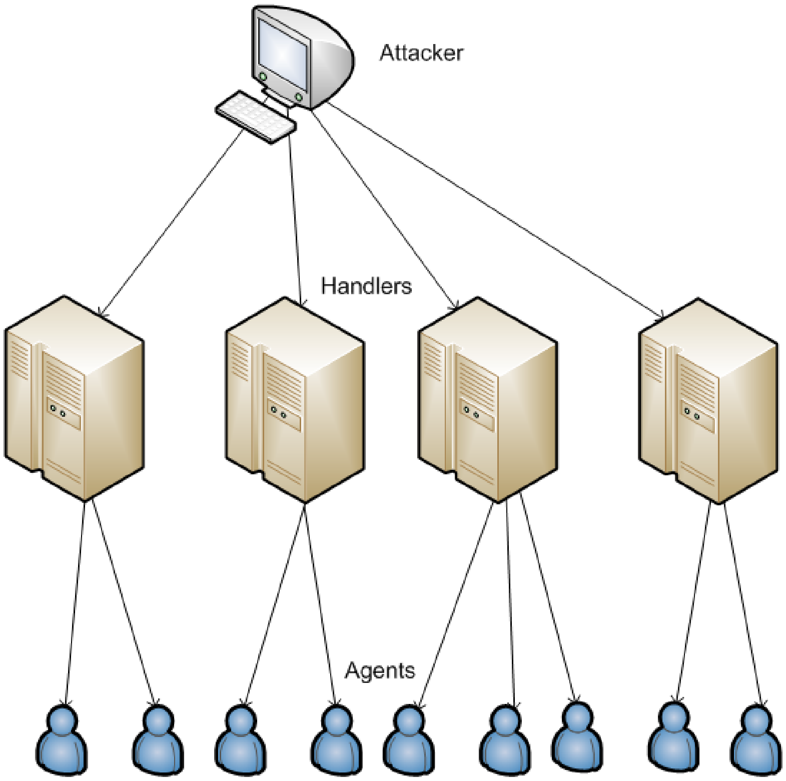


Figure 2

The communication between the attacker and the handler and between the handler and agents can be via TCP, UDP, or ICMP protocols. The owners and users of the agent systems typically have no knowledge that their system has been compromised and will be taking part in a DDoS attack. When participating in a DDoS attack, each agent program uses only a small amount of resources (both in memory and bandwidth), so that the users of these computers experience minimal change in performance (Lee & Specht, 2005).

1.2.1.2. IRC Based Agents

Internet Relay Chat (IRC) is a multi-user, on-line chatting system. It allows computer users to create two-party or multi-party interconnections and type messages in real time to each other. IRC network architectures consist of IRC servers that are located throughout the Internet with channels to communicate with each other across the Internet. IRC chat networks allow their users to create public, private and secret channels. Public channels are channels where multiple users can chat and share messages and files. Public channels allow users of the channel to see all the IRC names and messages of users in the channel. Private and secret channels are set up by users to communicate with only other designated users. Both private and secret channels protect the names and messages of users that are logged on from users who do not have access to the channel. Although the content of private channels is hidden, certain channel locator commands will allow users not on the channel to identify its existence, whereas secret channels are much harder to locate unless the user is a member of the channel.

An IRC-Based DDoS attack network is similar to the Agent-Handler DDoS attack model except that instead of using a handler program installed on a network server, an IRC communication channel is used to connect the client to the agents (see Figure 3). By making use of an IRC channel, attackers using this type of DDoS attack architecture have additional benefits. For example, attackers can use "legitimate" IRC ports for sending commands to the agents. This makes tracking the DDoS command packets much more difficult. Additionally, IRC servers tend to have large volumes of traffic making it easier for the attacker to hide his presence from a network administrator. A third advantage is that the attacker no longer needs to maintain a list of agents, since he can simply log on to the IRC server and see a list of all available agents. The agent software installed in the IRC network usually communicates to the IRC channel and notifies the attacker when the agent is up and running. A fourth advantage is that IRC networks also provide the benefit of easy file sharing. File sharing is one of the passive methods of agent code distribution. This makes it easier for attackers to secure secondary victims to participate in their attacks. A very common approach in connection is distributed clusters. In other words, each group of agents connects to an IRC server. When the attacker wants to communicate with all agents, he connects to all servers with expected zombies and issues the command. With this approach, it's much more difficult to track all zombies of a botnet and shut them down. A regulator would be able to track only zombies connected to the connected server and not all zombies (Lee & Specht, 2005).

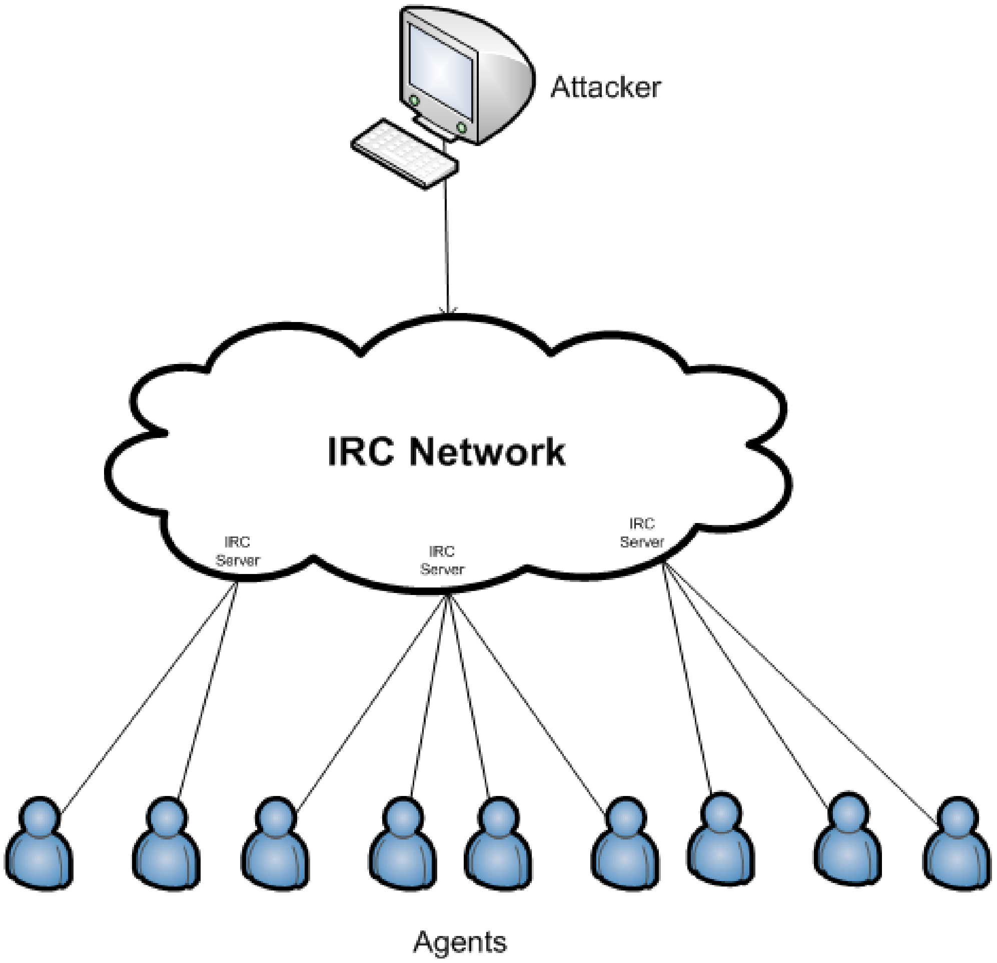
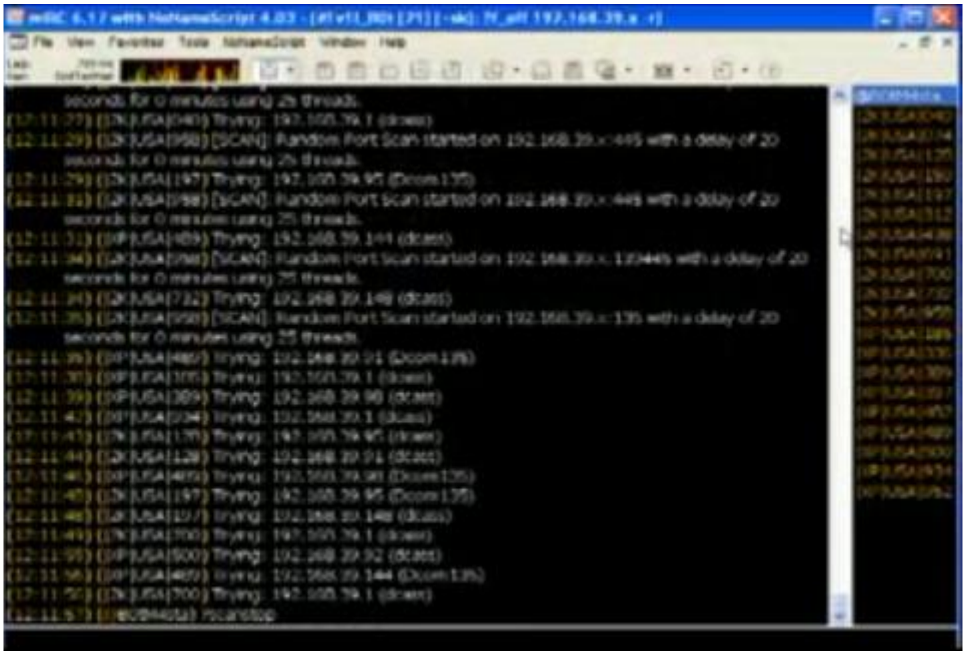


Figure 3

Figure 4 shows the interface of a very famous IRC program called mIRC. The opened window is a typical IRC channel. You can see bots logged in on the right hand side of the picture and they are sending keep-alive messages. In this scenario, the attacker would issue a command by changing the topic of the IRC channel to the give the green light for the bots joining the channel to start the action. The commands' syntax is written in a custom language that the attacker made and was embed in the bot software itself (YouTube, 2008).



mIRC 6.17 with NoNameScript 4.03 - [#1v11_l00t [71] [-sk]: !!_all 192.168.39.x -r]
File
View
Favorites
Tools
NoNameScript
Window
Help

```
        seconds for 0 minutes using 25 threads.
[12:11:27] [[2K|USA|040] Trying: 192.168.39.1 (dcass)
[12:11:29] [[2K|USA|958] [SCAN]: Random Port Scan started on 192.168.39.x:445 with a delay of 20
        seconds for 0 minutes using 25 threads.
[12:11:29] [[2K|USA|197] Trying: 192.168.39.95 (Dcom135)
[12:11:31] [[2K|USA|958] [SCAN]: Random Port Scan started on 192.168.39.x:445 with a delay of 20
        seconds for 0 minutes using 25 threads.
[12:11:31] [[XP|USA|489] Trying: 192.168.39.144 (dcass)
[12:11:34] [[2K|USA|958] [SCAN]: Random Port Scan started on 192.168.39.x:139445 with a delay of 20
        seconds for 0 minutes using 25 threads.
[12:11:34] [[2K|USA|732] Trying: 192.168.39.148 (dcass)
[12:11:35] [[2K|USA|958] [SCAN]: Random Port Scan started on 192.168.39.x:135 with a delay of 20
        seconds for 0 minutes using 25 threads.
[12:11:39] [[XP|USA|482] Trying: 192.168.39.91 (Dcom135)
[12:11:38] [[XP|USA|105] Trying: 192.168.39.1 (dcass)
[12:11:39] [[XP|USA|389] Trying: 192.168.39.98 (dcass)
[12:11:42] [[XP|USA|334] Trying: 192.168.39.1 (dcass)
[12:11:43] [[2K|USA|175] Trying: 192.168.39.95 (dcass)
[12:11:44] [[2K|USA|128] Trying: 192.168.39.91 (dcass)
[12:11:46] [[XP|USA|489] Trying: 192.168.39.98 (Dcom135)
[12:11:45] [[2K|USA|197] Trying: 192.168.39.95 (Dcom135)
[12:11:48] [[2K|USA|127] Trying: 192.168.39.148 (dcass)
[12:11:49] [[2K|USA|700] Trying: 192.168.39.1 (dcass)
[12:11:55] [[XP|USA|500] Trying: 192.168.39.92 (dcass)
[12:11:56] [[XP|USA|489] Trying: 192.168.39.144 (Dcom135)
[12:11:50] [[2K|USA|700] Trying: 192.168.39.1 (dcass)
[12:11:57] [[@USA|458] /scanstop
```

Figure 4

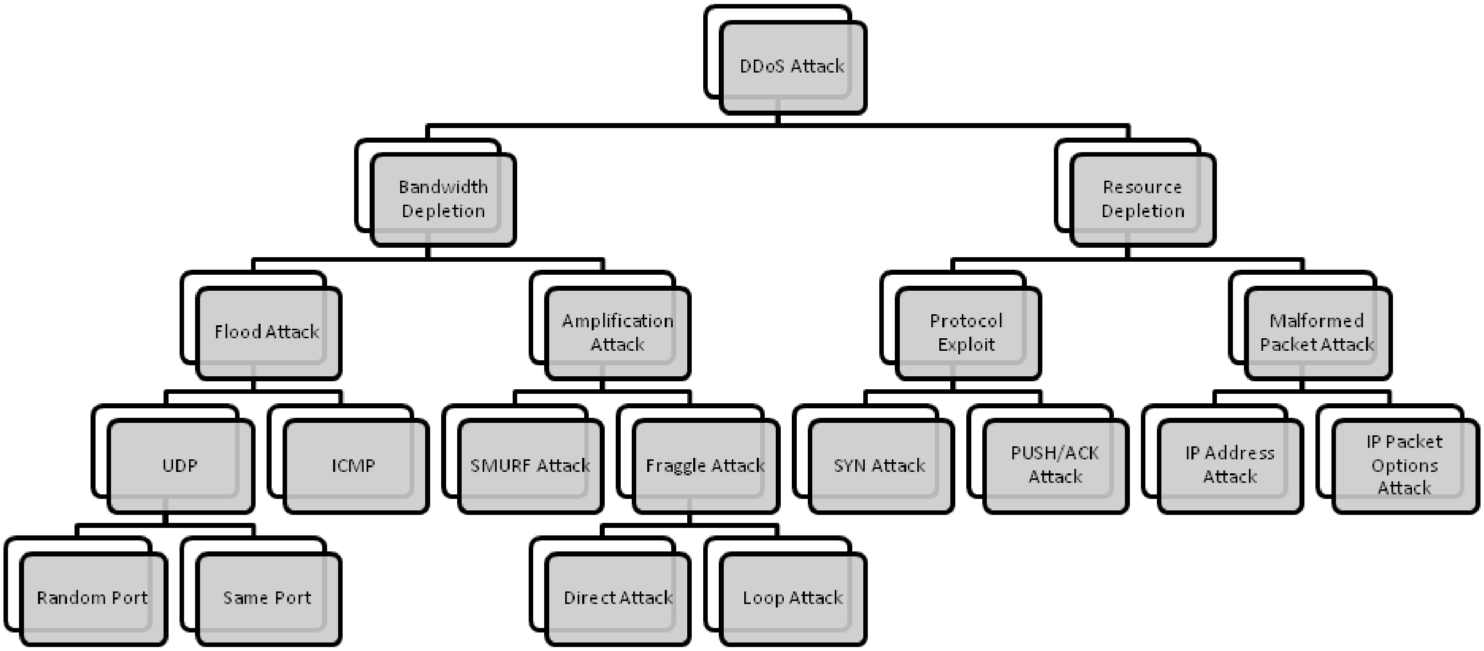


Figure 5

1.2.2. According to the exploit

Another classification for DDoS attacks is according to the exploit. What the attacker is going to exploit is a key point that affects the success or failure of the attack itself. Exploitation of vulnerabilities does not necessarily require strong technical background in some cases but in other cases the attacker must be a code guru who goes through assembly and reads data in Hex format! Figure 5 summarizes DDoS attacks' taxonomy.

1.2.2.1. Bandwidth depletion attacks

There are two main classes of DDoS bandwidth depletion attacks. A flood attack involves the zombies sending large volumes of traffic to a victim system, to congest the victim system's bandwidth. An amplification attack involves either the attacker or the zombies sending messages to a broadcast IP address, using this to cause all systems in the subnet reached by the broadcast address to send a message to the victim system. This method amplifies malicious traffic that reduces the victim system's bandwidth.

1.2.2.1.1. Flood attacks

In a DDoS flood attack the zombies flood the victim system with IP traffic. The large volume of packets sent by the zombies to the victim system slows it down, crashes the system or saturates the network bandwidth. This prevents legitimate users from accessing the victim. Figure 2 and Figure 3 indicate a flood attack for an Agent-Handler attack network and an IRC-based attack network respectively.

UDP Flood Attacks: User Datagram Protocol (UDP) is a connectionless protocol. When data packets are sent via UDP, there is no handshaking required between sender and receiver, and the receiving system will just receive packets and process it. A large number of UDP packets sent to a victim system can saturate the network, depleting the bandwidth available for legitimate service requests to the victim system.

In a DDoS UDP Flood attack, the UDP packets are sent to either random or specified ports on the victim system. Typically, UDP flood attacks are designed to attack random victim ports. This causes the victim system to process the incoming data to try to determine which applications have requested data. If the victim system is not running any applications on the targeted port, then the victim system will send out an ICMP packet to the sending system indicating a "destination port unreachable" message (Distributed Denial of Service: Trin00, Tribe Flood Network, Tribe Flood Network 2000, and Stacheldraht, 2000).

Often, the attacking DDoS tool will also spoof the source IP address of the attacking packets. This helps hide the identity of the secondary victims and it insures that return packets from the victim system are not sent back to the zombies, but to another computer with the spoofed address.

UDP flood attacks may also fill the bandwidth of connections located around the victim system (depending on the network architecture and line-speed). This can sometimes cause systems connected to a network near a victim system to experience problems with their connectivity.

ICMP Flood Attacks: Internet Control Message Protocol (ICMP) packets are designed for network management features such as locating network equipment and determining the number of hops or round-trip-time to get from the source location to the destination. For instance, ICMP_ECHO_REPLY packets ("ping") allow the user to send a request to a destination system and receive a response with the roundtrip time.

A DDoS ICMP flood attack occurs when the zombies send large volumes of ICMP_ECHO_REPLY packets to the victim system. These packets signal the victim system to reply and the combination of traffic saturates the bandwidth of the victim's network connection. As the UDP flood attack, ICMP attack source IP address may be spoofed as well (Lee & Specht, 2005).

1.2.2.1.2. Amplification Attacks

A DDoS amplification attack is aimed at using the broadcast IP address feature found on most routers to amplify and reflect the attack (see Figure 6). This feature allows a sending system to specify a broadcast IP address as the destination address rather than a specific address. This instructs the routers servicing the packets within the network to send them to all the IP addresses within the broadcast address range.

For this type of DDoS attack, the attacker can send the broadcast message directly, or the attacker can use the agents to send the broadcast message to increase the volume of attacking traffic. If the attacker decides to send the broadcast message directly, this attack provides the attacker with the ability to use the systems within the broadcast network as zombies without needing to infiltrate them or install any agent software. We further distinguish two types of amplification attacks, Smurf and Fraggle attacks.

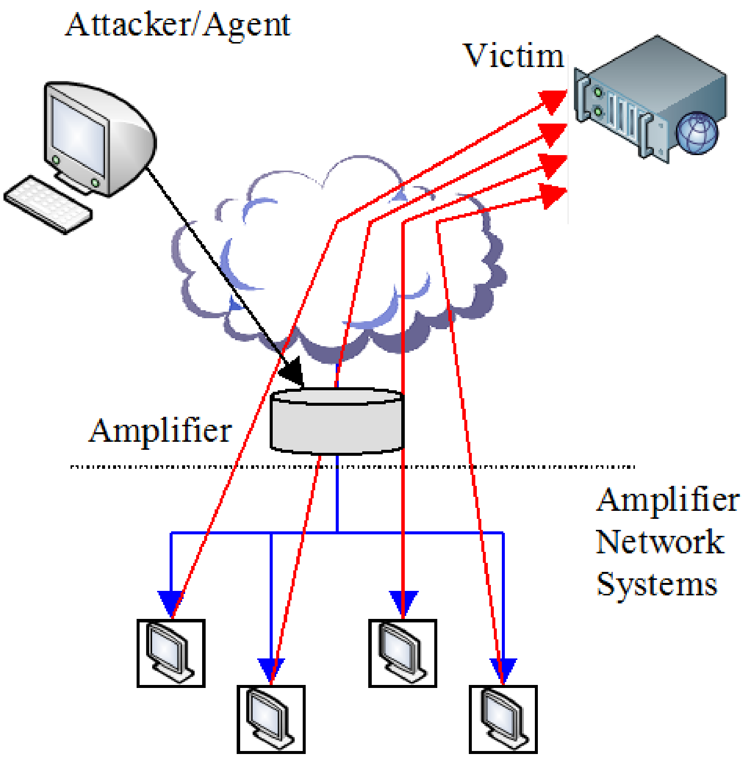


Figure 6 (Modified version from Lee & Specht, 2005**)**

Smurf Attacks: In a DDoS Smurf attack, the attacker sends packets to a network amplifier (a system supporting broadcast addressing), with the return address spoofed to the victim's IP address. The attacking packets are typically ICMP_ECHO_REQUESTs, which are packets (similar to a "ping") that request the receiver to generate an ICMP_ECHO_REPLY packet (TFreak, 2003). The amplifier sends the ICMP ECHO REQUEST packets to all of the systems within the broadcast address range, and each of these systems will return an ICMP ECHO REPLY to the target victim's IP address (Federal Computer Incident Response Center, 2000). This type of attack amplifies the original packet tens or hundreds of times.

Fraggle Attacks: A DDoS Fraggle attack is similar to a Smurf attack in that the attacker sends packets to a network amplifier. Fraggle is different from Smurf in that Fraggle uses UDP_ECHO packets instead of ICMP_ECHO packets (TFreak, fraggle.c, 2003). There is a variation of the Fraggle attack where the UDP_ECHO packets are sent to the port that supports character generation (chargen, port 19), with the return address spoofed to the victim's echo service (echo, port 7) creating an infinite loop (Martin, 2002). The UDP Fraggle packet will target the character generator in the systems reached by the broadcast address. Each of these systems generates a character to send to the echo service in the victim system, which will resend an echo packet back to the character generator, and the process repeats. This attack generates even more bad traffic and can create even more damaging effects than just a Smurf attack.

1.2.2.2. Resource depletion attacks

DDoS resource depletion attacks involve the attacker sending packets that misuse network protocol communications or sending malformed packets that tie up network resources so that none are left for legitimate users.

1.2.2.2.1. Protocol Exploit Attacks

TCP SYN Attacks: The Transfer Control Protocol (TCP) includes a full handshake between sender and receiver, before data packets are sent. The initiating system sends a SYN (Synchronize) request. The receiving system sends an ACK (acknowledgement) with its own SYN request. The sending system then sends back its own ACK and communication can begin between the two systems. If the receiving system is sent a SYNX packet but does not receive an ACKY+1 to the SYNY it sends back to the sender, the receiver will resend a new ACK + SYNY after some time has passed (Chen, 2000). The processor and memory resources at the receiving system are reserved for this TCP SYN request until a timeout occurs.

In a DDoS TCP SYN attack, the attacker instructs the zombies to send such bogus TCP SYN requests to a victim server in order to tie up the server's processor resources, and hence prevent the server from responding to legitimate requests. The TCP SYN attack exploits the three-way handshake between the sending system and the receiving system by sending large volumes of TCP SYN packets to the victim system with spoofed source IP addresses, so the victim system responds to a non-requesting system with the ACK+SYN. When a large volume of SYN requests are being processed by a server and none of the ACK+SYN responses are returned, the server begins to run out of processor and memory resources. Eventually, if the volume of TCP SYN attack requests is large and they continue over time, the victim system will run out of resources and be unable to respond to any legitimate users.

PUSH + ACK Attacks: In the TCP protocol, packets that are sent to a destination are buffered within the TCP stack and when the stack is full, the packets get sent on to the receiving system. However, the sender can request the receiving system to unload the contents of the buffer before the buffer becomes full by sending a packet with the PUSH bit set to one. PUSH is a one-bit flag within the TCP header (Defense Advanced Research Projects Agency, 1981). TCP stores incoming data in large blocks for passage on to the receiving system in order to minimize the processing overhead required by the receiving system each time it must unload a non-empty buffer.

The PUSH + ACK attack is similar to a TCP SYN attack in that its goal is to deplete the resources of the victim system. The attacking agents send TCP packets with the PUSH and ACK bits set to one. These packets instruct the victim system to unload all data in the TCP buffer (regardless of whether or not the buffer is full) and send an acknowledgement when complete. If this process is repeated with multiple agents, the receiving system cannot process the large volume of incoming packets and it will crash (Lee & Specht, 2005).

1.2.2.2.2. Malformed Packet Attacks

A malformed packet attack is an attack where the attacker instructs the zombies to send incorrectly formed IP packets to the victim system in order to crash the victim system. There are two types of malformed packet attacks. In an IP address attack, the packet contains the same source and destination IP addresses. This can confuse the operating system of the victim system and cause the victim system to crash. In an IP packet options attack, a malformed packet may randomize the optional fields within an IP packet and set all quality of service bits to one so that the victim system must use additional processing time to analyze the traffic. If this attack is multiplied using enough agents, it can shut down the processing ability of the victim system (Lee & Specht, 2005).

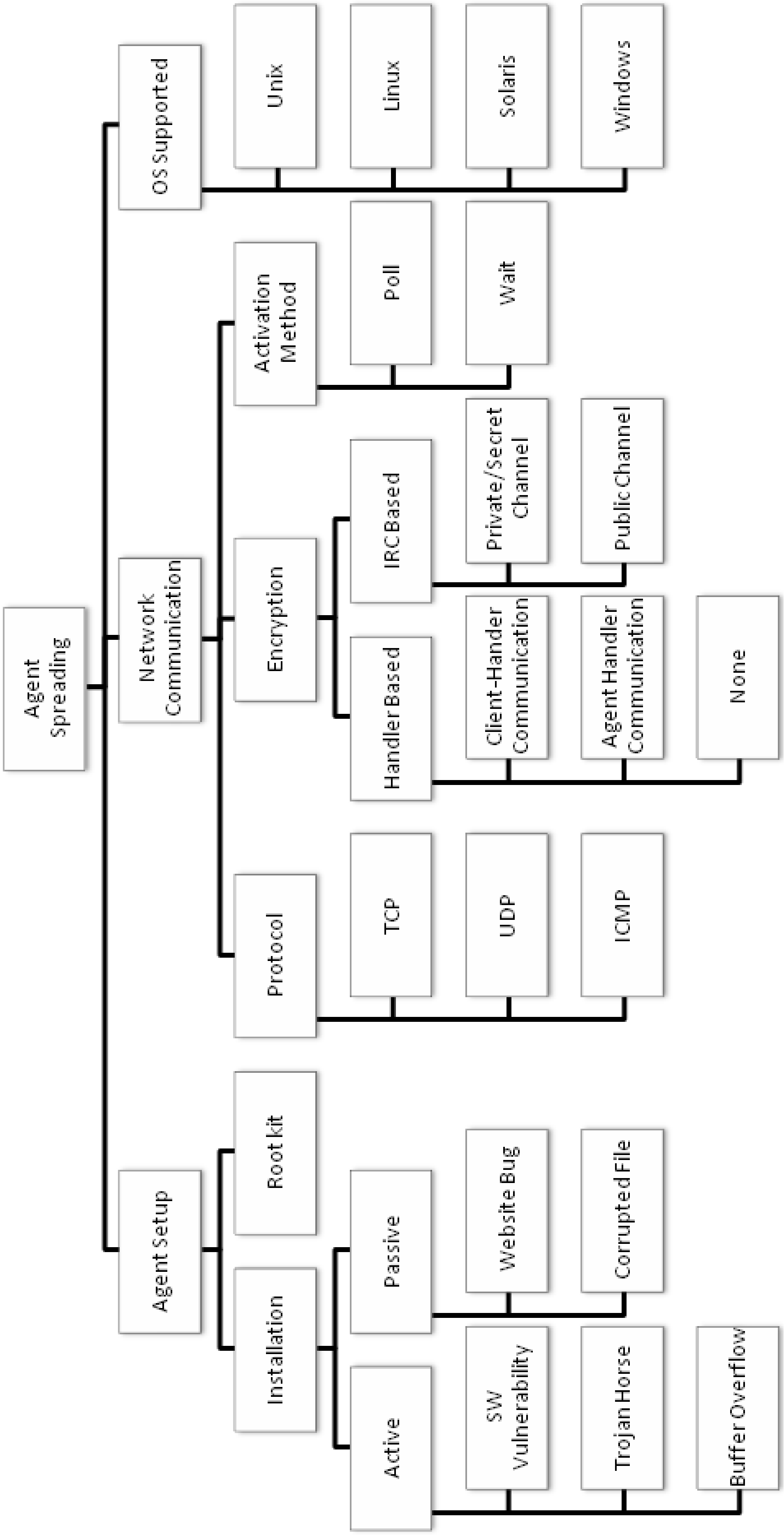
1.2.3. According to spreading technique

The previous two classifications showed clearly how severe DDoS attack might be. Yet a more clarifying classification should be mentioned. Classification according to the spreading technique is a classification that goes deep into the internal structure of the agents. It discusses how agents are set up on compromised computers, what methods of communication are used, and what platforms are targeted. Figure 7 summarizes the internal structure of agents.

1.2.3.1. DDoS Agent Setup

There are both active and passive methods that attackers use to install malicious code onto a secondary victim system in order to set up a Handler based or an IRC-based DDoS attack network. Active methods typically involve the attacker scanning the network for systems with known vulnerabilities. Upon identifying such vulnerable systems, the attacker runs scripts to break into the system. Once the attacker has broken into the system, he can stealthily install the DDoS Agent software. Thus the system is compromised as a secondary victim, and can be used as a zombie in a DDoS attack. Passive methods typically involve the attacker sharing corrupt files or building web sites that take advantage of known vulnerabilities in a secondary victim’s web browser. Upon accessing a file or website with an embedded DDoS Agent, the secondary victim system is compromised, and the DDoS agent code may be installed.

Figure 7


Agent Spreading
Agent Setup
Root kit
Installation
Passive
Website Bug
Corrupted File
Active
SW Vulnerability
Trojan Horse
Buffer Overflow
Network Communication
Protocol
TCP
UDP
ICMP
Encryption
Handler Based
Client-Hander Communication
Agent Handler Communication
None
IRC Based
Private/Secret Channel
Public Channel
Activation Method
Poll
Wait
OS Supported
Unix
Linux
Solaris
Windows

1.2.3.1.1. Active DDoS Installation

Scanning: Before launching a DDoS attack, attackers must first set up the DDoS attack network. They often run a scanning tool to identify potential secondary victim systems. One common tool attackers use to scan for ports is a software program called Nmap. Attackers can download Nmap from various locations on the web. This tool allows attackers to select ranges of IP addresses to scan. The tool will then proceed to search the Internet for each of these IP addresses. Nmap returns the information that each IP address is broadcasting such as TCP and UDP ports that are open, and the specific OS of the scanned system (Insecure.org, 2002). An attacker can then examine this list for potential secondary victim systems.

Another tool for scanning the network finds random IP addresses with a known vulnerability. This provides the attacker with a list of victim systems that all share the same common vulnerability. One example of this type of vulnerability scan tool is called Nessus (Nessus.org, 2002). Below, we describe three examples of vulnerabilities exploited for active DDoS agent setup.

Software/Backdoor Vulnerability: Once the attacker has scanned for a list of vulnerable systems, he will need to exploit the vulnerability to gain access to the secondary victim system and install the DDoS agent code. There are many sources on the Internet, such as the Common Vulnerabilities and Exposures (CVE) organization, which publicly list all the known vulnerabilities of different systems. CVE has currently categorized over 2,000 different types of vulnerabilities and they have over 2,000 more waiting for consideration (Homeland Security, 2002). This research information is available so network administrators can make their systems more secure; however, it also provides attackers with data about which vulnerabilities exist.

One such vulnerability was first reported in November of 2001 by Computer Emergency Response Team (**CERT**). The vulnerability reported was that the Kaiten IRC-Based DDoS agent software was being installed on Microsoft SQL Servers by making use of a known default password. Attackers could scan for hosts connected with TCP port 1433 (the MS SQL Server port), and find systems where they can then log on to MS SQL Servers using a default administrator password. From this administrator account, the attacker can utilize the “xp_cmdshell” procedure from the MS SQL Server to initiate an FTP session to download the agent software on the MS SQL Server (CERT Coordination Center, Carnegie Mellon Software Engineering Institute, 2001). Any MS SQL Server that has not had the administrator’s password changed or been upgraded with software patches to prevent the default administrator password could be victims of this type of attack.

Trojan Horse Program: A Trojan horse is a program that appears to perform a useful function, but in reality contains hidden code that either executes malicious acts or provides a trap door for unauthorized access to some privileged system function (Lee & Specht, 2005). Trojan horse programs are installed on a victim's system by the attacker and allow the attacker to gain control of a user's computer without the user knowing. In the case of a DDoS attack tool setup, Trojan horse programs already installed on a victim system might be used by the attacker to gain access to a secondary victim's system allowing the attacker to install the DDoS agent code. Trojan horse programs, themselves, are typically installed on a secondary victim's system by using the passive setup techniques discussed earlier.

Buffer Overflow: A buffer is a continuous block of memory (with a finite size) that serves as a temporary data storage area within a computer. A buffer overflow is an attack against the buffer that sends more data into the buffer than the size of the buffer. This causes the extra data to overwrite other information adjacent to the buffer (Cowan, Wagle, Pu, Beattie, & Walpole, 2000) in the memory storage stack, such as a procedure return address. This can cause the computer to return from a procedure call to malicious code included in the data that overwrites the buffer. This malicious code can be used to start a program of the attacker's choosing (such as a DDoS Agent) or provide access to the victim's computer so that the attacker can install the DDoS Agent code.

1.2.3.1.2. Passive DDoS Installation

Bugged Web Site: One method attackers can use to passively infiltrate a secondary victim computer system is to take advantage of a vulnerability found on web browsers. This method allows the attacker to create websites with code or commands to trap a victim. When the victim's web browser views the web page or tries to access content, the web page indirectly downloads or installs malicious code (e.g., a DDoS Agent.) One example of this type of attack exploits a bug in Microsoft's Internet Explorer (IE) versions 5.5 and 6.0. These versions of IE contain ActiveX, a technology developed by Microsoft to enable control within IE for viewing specific plug-in applications embedded within website code. ActiveX controls can be embedded within a website and allow the IE web browser to automatically download client binary code specified by the website being viewed (Microsoft, 1999).

An attacker can build malicious code into a web page that can take advantage of ActiveX. Instead of downloading client software for viewing the web page, the attacker can set up ActiveX to download a DDoS agent. The attacker will typically post a website with the malicious ActiveX code somewhere on the Internet to attract victim systems. The malicious html code is used to reference an ActiveX installation package that an IE browser will think is legitimate but actually contains code, such as DDoS agent software or code that allows the attacker to infiltrate the system. The malicious html code could include the DDoS agent, and takes the form:

```
<object classid=”clsid:XXXXXXXX” codebase= http://www.webpage.com/myactivex.cab> </object>
```

This command instructs IE to use an ActiveX control with a GUID (GUI Class Identifier) “XXXXXXXX” and if that control has not been downloaded, it provides the address where the control can be downloaded. The download address actually contains the malicious code the attacker wants to install on the victim machine. If the software patch for IE to prevent this problem is not installed, IE will inadvertently download the malicious code in place of the legitimate code. The attacker has now installed the malicious code on the victim’s computer. If this code is a DDoS agent, the attacker has now created a secondary victim system that can be invoked for a future DDoS attack. Actually this process creates an effect similar to that hacker in the movie “Untraceable” where each new visitor to the link actually becomes a zombie. So the more visitors, the more powerful the attack becomes.

Corrupted File: Another method of a passive attack that is commonly used is to alter files and include malicious code embedded within them. When the victim system tries to view or execute these files, they will become infected with the malicious code.

There are many tricks to creating infected files. Most attackers are skilled enough to embed a DDoS attack agent or other virus software within a legitimate file. The attackers redesign the desktop icons for such files, choosing long file names with legitimate extensions interwoven within the filename so that if only part of the file name is displayed, it will appear like a legitimate filename. For instance, one popular technique is for attackers to generate a text file with the binary executable code for a DDoS agent embedded within it. They rename the text file with a very long name with the .txt extension within the name when the real extension is .exe. For instance, the file might be newfile.txt_this_file_is_really_a_ddos_agent.exe. If only the first few characters of the file are displayed to the user, it will appear as if this file is really a text file, not an executable file. In this example, the newfile name would need to be around 150 characters long so most Windows systems would not show the full file name (Danchev, 2002). When a user launches the file, his machine will become infected with the DDoS agent software. Some attackers are skilled enough to include a text box to open, so the victim will think the file was legitimate and will not realize that it contained the DDoS agent.

Corrupt files can be exchanged in a variety of manners. Currently, IRC file sharing and Gnutella networks are two popular file-sharing methods that make it easy for a corrupt file to circulate to many users. An attacker can also send e-mail with corrupt files to victims, hoping the victims will open the files and infect themselves with the DDoS agent code.

1.2.3.1.3. Root kits

Root kits are programs that are used by the attacker after installation of handler and/or agent software to remove log files and any other records that might indicate that the attacker was using the system (Dittrich, 1999). Attackers may additionally use the root kit tools to create “backdoors" so that they will be able to access the secondary victim's system in the future. Root kit tools are typically used when handler software is installed since one handler can be critical for the DDoS attack network to work and since handler programs are usually installed within ISP or corporate networks where the possibility of detection may be higher. In comparison, the effort to use a root kit on all of the agents may be time prohibitive and less important since secondary victims are less likely to be aware of the agent software and if some of the agents are discovered, their loss does not significantly impact the DDoS attack network (Lee & Specht, 2005).

1.2.3.2. Network communications

There are several aspects in communications between the different components of a botnet that we want to investigate; Protocols used, encryption schemas and activation methods.

1.2.3.2.1. Protocols

The DDoS agents and handlers can communicate to each other via TCP, UDP, and/or ICMP. DDoS handlers and clients can also communicate with each other using the same protocol options.

1.2.3.2.2. Encryption

Some DDoS attack tools have also been developed with support for encrypted communication within the DDoS attack network. Handler based DDoS attacks might use an encrypted channel either between the client and the handlers, or between the handlers and the agents. The method of encryption for agent-handler DDoS attacks are dependent on the communication protocol used by the DDoS tool. IRC-based DDoS attacks may use either a public, private, or secret channel to communicate between the agents and the handlers. Both private and secret IRC channels provide encryption, however private channels (not the data or users) appear in the IRC server's channel list and secret channels do not appear in the IRC server's channel list.

1.2.3.2.3. Activation Methods

There are two key methods for the DDoS agents to be activated. In some DDoS tools, the agents actively poll the handlers or IRC channel for instructions, whereas in other DDoS tools, the agents will lie and wait for communication from either the handler or the IRC channel

1.2.3.3. Supported operating systems

DDoS attack tools are typically designed to be compatible with different operating systems (OS). Any OS system (such as UNIX, Linux, Solaris, or Windows) may have DDoS agent or handler code designed to work on it. Typically, the handler code is designed to support an OS that would be located on a server or workstation at either a corporate or ISP site. This usually leads to the choice of UNIX, Linux, or Solaris in the Handler model. However, recently window’s boxes are spreading rapidly forcing the attackers to write code to utilize the massive existing infrastructure running Microsoft software. For the agent code, it is also common for it to be compatible with Linux or Solaris with the addition of Windows. Many attackers target residential Internet users with DSL and cable modems (for higher attacking bandwidth) and these users typically use Windows.

1.3. Known Incidents

After digging deep into technical issues of botnets and DDoS attacks, it’s important to list some of the major real world attacks that have taken place. These incidents clarify very important key points that concerns economical, political and privacy issues.

1.3.1. DDoS attack on root name servers

DDoS attacks on root name servers are like a nuclear attack in the real world. But this one is in the cyber space. The attacker targets one or more of the thirteen DNS root servers. These attacks are extremely significant, as the root name servers function as the Internet backbone, translating text-based Internet hostnames into IP addresses. As the name servers provide this service for DNS lookups worldwide, attacks against the root name servers are attempts to disable the Internet itself, rather than specific websites (Wikipedia, 2009).

An attack occurred on February 6, 2007 at 10:30 UTC, and lasted about five hours. Although none of the servers crashed, two of the root servers reportedly "suffered badly", while others saw "heavy traffic". The botnet responsible for the attack has reportedly been traced to the Asia-Pacific region. There was some speculation in the press that the attack originated from South Korea (Wikipedia, 2009).

1.3.2. Russia DDoS attack on Estonia

This attack is rather a political attack. It has a story. Once upon a time, when Estonia was part of the Soviet Union, they had this Soviet Red Army statue that resembled victory for the Soviets and misery for the Estonians. In January 2007, the president of Estonia has signed into law a bill allowing the removal of the controversial Soviet war memorial from the centre of the capital Tallinn (BBC News, 2007). The Russians thought that was inappropriate and warned the Estonians about the consiquences (TIMES Online, 2007). Having suffered a lot and looking forward to independence, the Estonians ignored the threats and actually removed the statue.

This was the ignition to the cyber war that Russia drove over Estonia in 2007. Although the Russian government officially doesn’t have a hand at that, all traffic was traced back to locations in Russia. The attacks turned down e-banking websites, governmental portals, and critical communications websites for more than a week. Almost the whole backbone of the country was disabled (Guardian, 2007). Richard A. Clark - counter-terrorism adviser on the U.S. National Security Council - was talking in BOSTON security conference 2008 about this issue and how dangerous it is to be exposed that much. We have too much information on the internet that life can be paralyzed with these kinds of attacks, he said (Clark, 2008).

1.3.3. BBC Experiment

“An investigation by the BBC into cybercrime may itself have broken UK computer crime law”

This was the headline for “The register” magazine on the 12th of March 2009. You can imagine how terrifying that was. BBC Click, a program covering news and recent developments in the world of consumer technology, got its hands on a botnet of 22,000 compromised PCs from an underground forum! It used these machines to send spam to two accounts it had established with Gmail and Hotmail. The program also used these zombie machines to show how they might be used in a denial of service attack. After getting permission from security firm Prevx, which commented on camera but did not otherwise participate in the investigation, BBC Click used the compromised machines to flood a backup site run by the security firm with junk traffic.BBC Click found that only 60 compromised machines were needed to render Prevx's site inaccessible. The broadcaster then warned the owners of the infected computers that their machines were compromised, and advised on how to clean them, by changing their screensaver (Leyden, 2009). All this was recorded and played on TV!

This experiment not only shows how easy it is for an evil attacker with bad intentions to get his hand on a botnet, but also provides us with a vision. It’s getting pretty grey here. The right and wrong actions are not very clear. A lot of law enforcement firms raised questions about the legality of this experiment leaving us with only questions but no answers.

2. DDoS Prevention

Many institutes and teams are researching about DDoS and how we can prevent it. However, it was not an interest for the people in charge to spend more money and effort on that issue. Now, after the latest events, it’s very clear that we are in great need for taking decisions to try to reduce that risk as much as possible, hoping to prevent and/or contain the problem. There are currently few procedures/efforts to reduce DDoS attacks that are very local and ineffective in case of big hits. Yet a strong research wave is going on trying to find a feasible solution. We discuss these efforts and suggest some of the new techniques that if implemented correctly can prevent DDoS attacks.

2.1. Current efforts to thwart DDoS attacks

Current efforts are very local as mentioned earlier because there’s no coordination between various authorities. The current model is a distributed authority model. Each ISP, country has its own set of rules/laws that govern cyber security. In some countries even there’s no authority at all. Each user is responsible for his actions and that’s it! It’s known that distributed systems in general are more complicated than centralized systems. Although real world is distributed, one can argue that any aspect of life have to be centralized to come to an order.

2.1.1. ISP Filtering

Almost all ISPs have IP filters to try preventing various types of attacks. This actually helps very much in preventing some of the attacks like the SMURF attacks mentioned earlier. Because IP Address spoofing is not possible with IP filtering in action. Also a lot of ISPs detect malicious packets with known signatures and/or bad TCP/UDP options. This also prevents few attack vectors. Also governments can enforce policies on some ISPs to block certain routes and/or subnets in case of attacks, political issues…etc. But these efforts are very local even ISPs in the same country often do not share data about blacklisted IPs or malicious hosts.

2.1.2. Monitoring Teams

There are several teams that monitor internet activity and create a dynamic list of IPs with a metric for their threat level. Team Cymru, as an example, monitors specific Internet critical infrastructure, providing the results in this section. This permits the viewer to determine the scope and duration of Internet-effecting outages, and the localized effect of such outages. Many such monitoring projects use ICMP (ping), yet this isn't a great measure of performance. For this reason we also monitor connectivity to Internet critical infrastructure and between Team Cymru pods using both TCP and UDP. The monitoring focuses on DNS and BGP, the two most critical services. The DNS monitoring includes both the DNS service as well as network connectivity to the given name server. The BGP monitoring is based on peering with over 100 BGP-speaking routers, providing us with a granular view of the internal routing tables (Team Cymru, 2009).

With team Cymru monitoring one can create a list of “bad” IPs or IPs that are possibly part of botnets. This is very useful information. However, unfortunately, not used efficiently.

2.1.3. Emergency Data Centers

A common practice for large enterprises is having disaster recovery datacenters and/or emergency datacenters. There were people arguing about this method as a mitigation technique. This is actually not a mitigation technique from my point of view. It’s something like tranquilizers that just make you live a little bit longer! Although it is necessary, yet, it must be combined with real mitigation and preventions techniques.

2.1.4. Host Hardening

Host based security software is becoming very popular. In fact, this is a good thing and a bad thing at the same time. Having a security suite installed on the PC, a simple user might think that he’s completely safe. This is totally untrue. Without aggressive updates, anti malware is almost useless. Even with updated anti malware software, new attacks (a.k.a. 0days) are very hard to catch and might evade all filters. As for our concern, Host hardening reduces the possibility of being part of a botnet. So, we have to increase the awareness the users about how to harden their personal computers and appliances not just by installing anti malware, but also by following best practices and avoiding falling into common tricks that might get them into trouble.

An example of a very tricky exploit happened at North Dakota in Q1 2009, a US state. Hybrid cars in North Dakota have been tagged with fake parking citations that include a Web address hosting malicious software that drops a Trojan onto the computer. The yellow tickets found on the cars in Grand Forks, North Dakota, read "PARKING VIOLATION. This vehicle is in violation of standard parking regulations. To view pictures with information about your parking preferences, go to" and gave a Web site! (Cnet News, 2009). This shows that users actually might get fooled because that’s the human’s nature. “Only two things are infinite, the universe and human stupidity, and I'm not sure about the former. (Albert Einstein *1879*) ”. So we need awareness again.

2.1.5. Hunting Tools

There are several tools for hunting botnets that are not publicly available used by some law enforcement authorities. Microsoft has developed one of those tools recently. Although Microsoft is reluctant to give out details on its botnet buster -the company said that even revealing its name could give cyber criminals a clue on how to thwart it!- company executives discussed it at a closed door conference held for law enforcement professionals. The tool includes data and software that helps law enforcers get a better picture of the data being provided by Microsoft's users, said Tim Cranton, associate general counsel with Microsoft's World Wide Internet Safety Programs. "I think of it ... as botnet intelligence," he said (PCworld, 2008).

2.2. Suggested methods to thwart/prevent DDoS attacks

All the previous methods were not very effective. We've already seen too much incidents in the past couple of years that we can say we are in need for innovative solutions; Solutions that depends on cooperation between different authorities to unify efforts against the evolving beasts (botnets).

2.2.1. Internet Interpol

This term is not a valid term now. But I think it might be the solution for the DDoS problem. There exists a task force in the Interpol that is concerned with cyber malicious activities, but it is not doing any active work on its own. They are just a follow up or support teams for the tactical physical operations. Having a centralized international task force that is regulating the internet, forcing standards, coordinating between countries "cyber political" issues, if I may call it, would be the dawn of a new age. For the same reasons that the Interpol was constructed, there should be an internet Interpol. Decision makers usually fear the word "regulation". It means to them privacy nightmares. In fact it is, if used in a bad way. However, given the current situation of modern threats, it has become a must. As transparent as possible and less breaching as possible, regulation will solve the problem gradually. There are key points that are to be discussed in this approach to ensure the efficiency and transparency of this program.

2.2.1.1. Scope

A very important point to discuss is the scope of this task force; internet Interpol. It's preferred that internet Interpol to be concerned with all cyber security threats. However we are only concerned with DDoS attacks and botnets so we will focus our study on them.

Internet Interpol should be connected to existing Interpol authorities. This ensures a smooth introduction to this task force. Also, having access to the information that Interpol already have, makes the job easier for the analysts. And, of course, having the same managerial structure saves a lot of time doing background checks, business analysis and all other legal issues that should be investigated first before devising such a new task force.

2.2.1.2. Connection

Internet Interpol must be connected to all ISPs. This ensures the orchestration of communications between ISPs internally (inside the same country) and externally (across countries) to enforce policies, share information about high threat IPs / subnets, and aggregate statistics to create estimated mathematical and statistical models. Being connected to all ISPs it can create a reputation system based on statistics collected from various locations.

2.2.1.3. Role

As mentioned earlier, internet Interpol will be having a lot of tasks and missions. However, we are only concerned about its role as a prevention system for DDoS attacks.

Building the reputation system is one of the main countermeasures that internet Interpol will utilize. This reputation system flags each IP with a threat level which indicates the probability that this IP is part of a botnet. IPs above a certain threshold are considered a high risk. There are several models for dealing with high risk hosts.

A signal can be sent over to the host ISP to block it way up the routing chain. As opposed to current blocking system where the traffic is blocked at the end of the chain where the attack is detected.

Figure 8 shows the original method currently used by ISPs. Due to lack of communication between ISPs, the bad traffic actually was routed from the zombie all the way over the route till the last ISP in the chain. This ISP (ISP 2) detected that the server is under a DDoS attack and by maintaining its own IP blacklist, its firewall blocked the traffic. This scenario is way far from perfect.

On the contrary, as we see in Figure 9, when the internet Interpol detected that ISP 2 is under attack, it dynamically communicated with the originating ISP (ISP 1) and blocked the traffic coming from the hosts with high threat level.

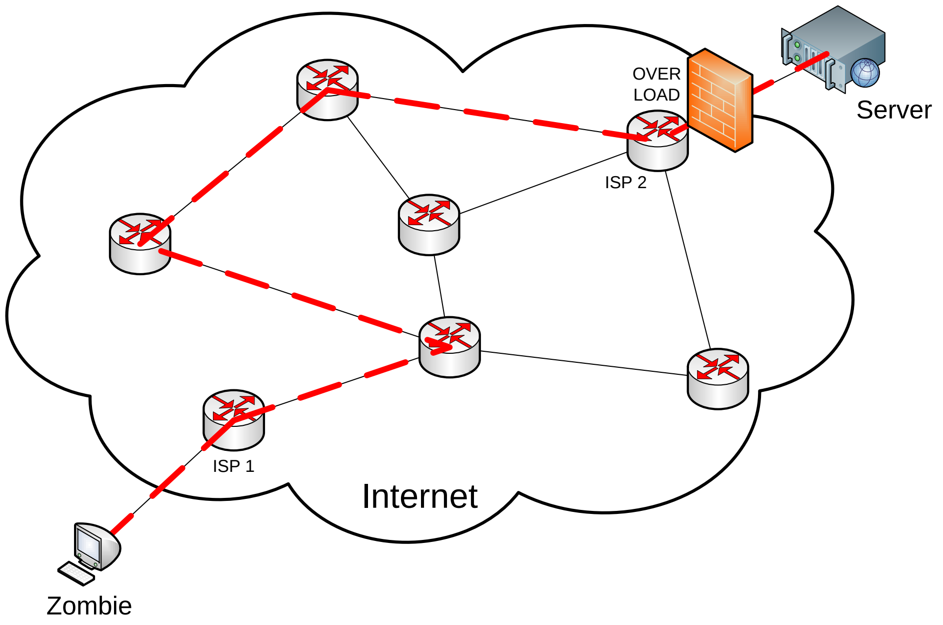


Figure 8

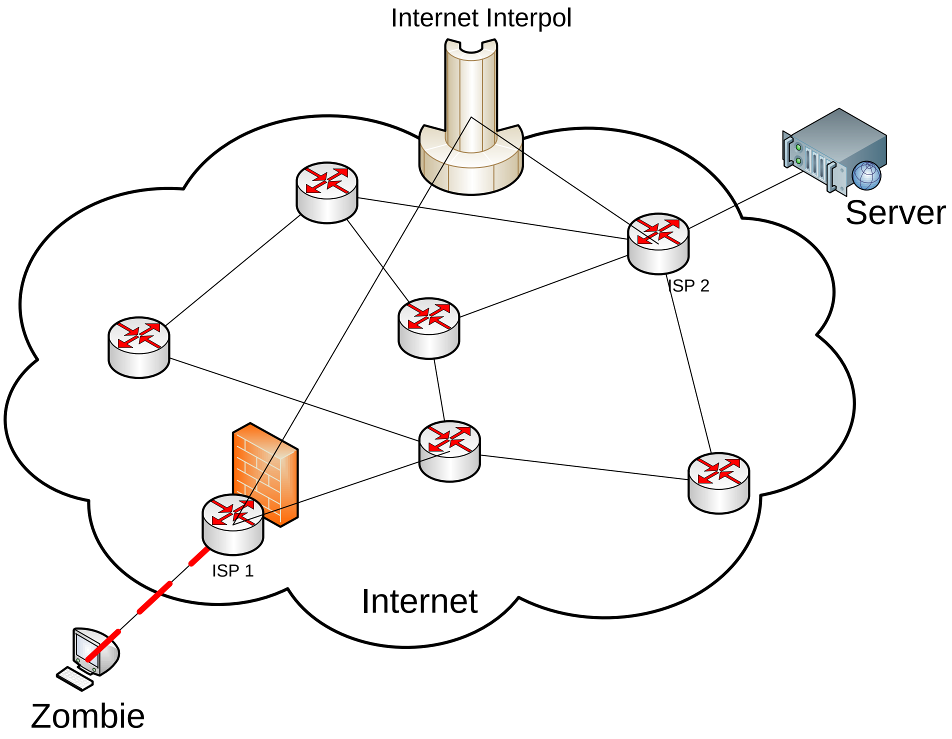


Figure 9

There’s a very important issue though, in the previous model. The user that is being blocked is actually an innocent person! The user must be warned that his IP is being blocked for certain security reasons. One can imagine a scenario where the user is directed to a website hosted by the internet Interpol. This site informs the user exactly about the reason for the black listing and offers a method for removing the blacklist. An online scan can be a good solution. After the user get black listed, he must remove the malware from his computer on his own. And when the computer is clean, he goes to the internet Interpol website to do an online scan. The results of the scan will give the green light to the ISP to remove the IP from the black list. Another possibility is to offer an online scan and removal. This can be charged or free of charge according to the business model for the firm itself. Anyway, this is a point that should be investigated from the business and legal points of view.

Also dynamic blocking has to take care about the dynamic IP Addresses. Usually ISPs provide residential users with dynamic IP addresses and does not allocate static IPs. ISPs should send a signal to the internet Interpol whenever an IP is changed to either update the threat levels as necessary, migrate the profile of the user from the old IP to the new IP, or delete the old profile and start all over again.

There are certain legal problems that have to be studied as well concerning who has jurisdiction in case of attacks and how to handle all that. But that is out of the scope of this research.

2.2.2. Heterogeneity in operating systems

Consider this scenario. There is only one operating system in the world that everybody uses. Now it’s going to be very easy for the attackers to write one exploit that runs on every single machine on earth! On the contrary, if every single machine had its own operating system, then an attacker must write malware for every specific user. In this latter scenario, botnets would not exist for sure. It would require a gigantic amount of work.

The point from this argument is that heterogeneity of platforms makes it statistically harder on the attacker to write a malware that spreads well. The problem is that most of the personal computers on earth run Microsoft software. Recently, servers also are migrating to Microsoft. This fact makes the decision pretty easy for the attacker when he is choosing the platform under which his agents are going to work. It’s very healthy for the whole internet to have some sort of balance between operating systems on both client and server sides to make the job harder for the attackers.

Internet Interpol cannot help in this situation. However, national mass communications methods can publicize this culture on TVs and awareness could be spread to increase the security knowledge of the public.

2.2.3. Cloud computing

Cloud computing has been out there for while now. It is actually doing quite well. Amazon EC2 and Google clouds are getting pretty big and are used in numerous ways. Unfortunately, cloud computing is used for bad purposes also. Phishers are using cloud endpoints to provide their network with load balancing and survivability. Fast-flux enabled phishing sites using rapid DNS rotation across a large number of end points helps phishers evade most filters.

Are you thinking what I'm thinking? Why not use the same technique to provide survivability against DDoS attacks. With backup websites on the cloud and a good plan for the rotation, one could make use of multiple small servers across multiple cloud vendors and survive a strong DDoS attack. This method has to be tried for scalability and robustness and require a lot of funding which is beyond the scope of this research. Yet, it remains a very promising solution.

2.2.4. Project: The Citadel

We've seen the political cyber war happening already. As mentioned earlier, Estonia lost a lot in that war back in 2007. No solid numbers but it was a great loss. On the national level, there should be a cyber seal plan. Richard A. Clarke talked about that in SOURCE Boston conference 2008. China is well prepared to seal off the entire country in case of cyber war, he said (Clark, 2008). But in fact, china is the ONLY country that is prepared for something like that. This is very critical matter; A matter of national security. Every country should have its own plan to seal off its cyber boarders in case of cyber war, and continue delivering local electronic services.

2.2.5. Law Enforcement

Although this research is a technical research, it's worth mentioning that not only the technical side is the issue here. All what we have been talking about to prevent DDoS attacks has nothing to do with the real criminals themselves. Cyber crimes are very easy to commit and very tempting, because the punishment is not imminent. Usually, when attackers are doing their attacks they have this feeling that they are safe, because they are sitting at home or at a cyber café physically way far from the attack. Why is cybercrime numbers raging while physical crimes are coming to a settlement? The answer of this question cannot be answered directly. However, one can argue that usually cyber criminals do not have a visualization of what can happen to them if they were caught and usually they think that they will not get caught at all!

We have to have more strict laws that define cyber crimes and its penalties. Some countries have some laws, others have few laws and others do not have laws concerning cybercrime at all! Even countries that have laws, do not spend much effort on tracking and hunting down attackers. Also trials of cyber crimes have to be more publicized. People have to know that playing around the cyberspace is not a game anymore and there are strict laws that are well applied.

The United States is taking good steps in that direction. We have John Schiefer, a botmaster, sentenced for four years prison and fined $20,000 and earlier in 2007. Also Microsoft has announced in February 2009 that it is offering a reward of $250,000 to anyone who can provide information that can help arrest the creator of the Conficker worm (i.e. the botmaster of the Conficker botnet). “Microsoft’s reward offer stems from the company’s recognition that the Conficker worm is a criminal attack,” a Microsoft statement said (Mail Online, 2009). This is actually a good start. But In order to teach the attackers a lesson, we need this spirit to propagate to the Far East, Middle East and Russia as well.

3. Conclusion

All this leads us back to the picture that we imagined earlier at the beginning of this section and thought might never happen or at least far from us. “This page cannot be displayed” is a scenario that is not science fiction anymore. Without serious efforts and detailed research to thwart these crawling monsters, we will find ourselves disconnected and back to ages were we used to go the post office to send a mail! Of course there is no silver bullet in security. Nothing is totally secure. However, with the proposed techniques, the risks are reduced to a much lower limit than it is currently.

With internet Interpol in place, we have a regulated internet. Heterogeneous platforms do the statistical magic. Cloud computing is a good backup system. Having a seal off plane B in case of fatal damage is very important. And fair applied laws ensure that not only people get their rights but also evil intended people fearing to commit crimes because they won’t get away unpunished. This whole system if assembled together can definitely thwart these crawling monsters.

4. References

BBC News. (2007, Jan). *Estonia to remove Soviet memorial* . Retrieved from http://news.bbc.co.uk/2/hi/europe/6255051.stm

BBC News. (2008, Sep). *Zombie plague sweeps the internet*. Retrieved from http://news.bbc.co.uk/2/hi/technology/7596676.stm

CERT Coordination Center, Carnegie Mellon Software Engineering Institute. (2001, Nov). *CERT Incident Note IN-2001-13*. Retrieved from Cert.org: http://www.cert.org/advisories/CA-2001-20.html

Chen, Y. W. (2000). *Study on the Prevention of SYN Flooding by Using Traffic Policing*. Retrieved from Network Operations and Management Symposium, 2000. NOMS 2000. 2000 IEEE/IFIP: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=830416

Clark, R. A. (2008). *The Current State of the War on Terrorism and What it Means for Homeland Security and Technology*. Retrieved from SOURCE Boston Conference: http://www.sourceconference.com/index.php/pastevents/source-boston-2008-videos

Cnet News. (2009, Feb). *Fake parking tickets direct to malicious Web site*. Retrieved from http://news.cnet.com/8301-1009_3-10156841-83.html

Cowan, C., Wagle, P., Pu, C., Beattie, S., & Walpole, J. (2000). *Buffer Overflows:Attacks and Defenses for the Vulnerability of the Decade*. Retrieved from Oregon Graduate Institute of Science & Technology: www.ece.cmu.edu/~adrian/630-f04/readings/cowan-vulnerability.pdf

Danchev, D. (2002, Oct). *The Complete Windows Trojans Paper*. Retrieved from BCVG Network Security: http://www.ebcvg.com/articles.php?id=91

Defense Advanced Research Projects Agency. (1981, Sep). *Transmission Control Protocol DARPA Internet Program Protocol Specification (RFC 793)*. Retrieved from IEFT.ORG: http://www.ietf.org/rfc/rfc0793.txt

Distributed Denial of Service: Trin00, Tribe Flood Network, Tribe Flood Network 2000, and Stacheldraht. (2000, Feb). CIAC: U.S. Department of Energy.

Dittrich, D. (1999, Oct). *The DoS Project's "Trinoo" Distributed Denial of Service Attack Tool*. Retrieved from http://staff.washington.edu/dittrich/misc/trinoo.analysis

Federal Computer Incident Response Center. (2000). *Defense Tactics for Distributed Denial of Service Attacks.*

Homeland Security. (2002, Mar). *Common Vulnerabilities and Exposures (CVE)*. Retrieved from cve.mitre.org: http://cve.mitre.org/

Information Security Magazine. (2006, Jul). *Distributed Denial of Service Attack*. Retrieved from http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci557336,00.html

Insecure.org. (2002, Aug). *Nmap Stealth Port Scanner Introduction*. Retrieved from http://www.insecure.org/nmap/

Lee, R. B., & Specht, S. M. (2005, Jan). *Taxonomies of Distributed Denial of Service Networks, Attacks, Tools and Counter Measures.* Retrieved from Prinecton University.

Leyden, J. (2009, Mar). *BBC botnet investigation turns hacks into hackers*. Retrieved from TheRegister.co.uk: http://www.theregister.co.uk/2009/03/12/bbc_botnet_probe/

Mail Online. (2009, Feb). *Microsoft announce $250,000 bounty on worm creator*. Retrieved from http://www.dailymail.co.uk/sciencetech/article-1144542/Microsoft-announce-250-000-bounty-catch-worm-creator.html

Martin, M. J. (2002, Oct). *Smurf/Fraggle Attack Defense Using SACLS*. Retrieved from www.searchnetwork.techtarget.com: http://searchnetworking.techtarget.com/tip/1,289483,sid7_gci856112,00.html

Microsoft. (1999, Jun). *How to Write Active X Controls for Microsoft Windows CE2.1*. Retrieved from http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnce21/html/activexce.asp

Nessus.org. (2002). *Nessus Documentation*. Retrieved from http://www.nessus.org/

PCworld. (2008, Apr). *Microsoft Botnet-hunting Tool Helps Bust Hackers*. Retrieved from http://pcworld.about.com/od/securit1/Microsoft-Botnet-hunting-Tool.htm

Team Cymru. (2009). *Team Cymru Monitoring*. Retrieved from http://www.team-cymru.org/Monitoring/

TFreak. (2003). *fraggle.c*. Retrieved from phreak.org: http://www.phreak.org/archives/exploits/denial/fraggle.c

TFreak. (2003, May). *Smurf.c*. Retrieved from Phreak.org: http://www.phreak.org/archives/exploits/denial/smurf.c

The Guardian. (2007, May). *Russia accused of unleashing cyberwar to disable Estonia*. Retrieved from http://www.guardian.co.uk/world/2007/may/17/topstories3.russia

TIMES Online. (2007, Apr). *Russia threatens Estonia over removal of Red Army statue*. Retrieved from http://www.timesonline.co.uk/tol/news/world/europe/article1714401.ece

Wikipedia. (2008, Feb). *Botnets*. Retrieved from http://en.wikipedia.org/wiki/Botnet

Wikipedia. (2009, Jan). *DDoS Attacks on Root Name Servers*. Retrieved from http://en.wikipedia.org/wiki/DNS_Backbone_DDoS_Attacks

YouTube. (2008, Apr). *Building Botnets*. Retrieved from http://www.youtube.com/watch?v=C56ulcvYRE8