SECURING APACHE : DOS & DDOS ATTACKS - II

How DDoS attacks are performed

A DDoS attack has to be carefully prepared by the attackers. They first recruit the zombie army, by looking for vulnerable machines, then break into them (usually via Metasploit), and install attack toolkits. The attacker next establishes communication channels between machines, so that they can be controlled and engaged in a coordinated manner.

This is done using either a handler/zombie architecture, or an IRC-based command-and-control (C&C) channel. Once the DDoS network is built, it can be used to attack various targets as often as desired.

But, the first step for attackers is to find machines that they can compromise. To maximise the yield, they will recruit machines that have good connectivity and ample resources, and are poorly maintained. Unfortunately, many of these exist within the pool of the millions of Internet hosts.

The process of looking for vulnerable machines is called scanning. The attackers send a few packets to the system, to see whether it is alive (online) and vulnerable. If so, the attackers will attempt to break into the machine. After breaking in, they will try to install malicious code, or a “bot” on the machine, which will give them complete reverse control over the machine. The attacker can repeat the scan-and-infect process to build big attack networks.

The mechanism of propagating malicious code to compromised machines is classified into three types. These are given below.

Central source propagation

In this method, after discovering a vulnerable system that will become one of the zombies, the attackers take control over the system and download an attack toolkit from a different website (a central source) to the newly exploited system. After the toolkit is transferred, an automatic installation of the attack tools takes place on this system. That initiates a new attack cycle, where the newly infected system looks for other vulnerable computers on which it can install the attack toolkit, using the same process as the attackers.

Like other file-transfer mechanisms, this mechanism commonly uses HTTP, FTP, and RPC (remote-procedure call) protocols. From the attacker’s perspective, such central sources can be easily identified and disabled by security professionals; hence, this method is not used much nowadays.

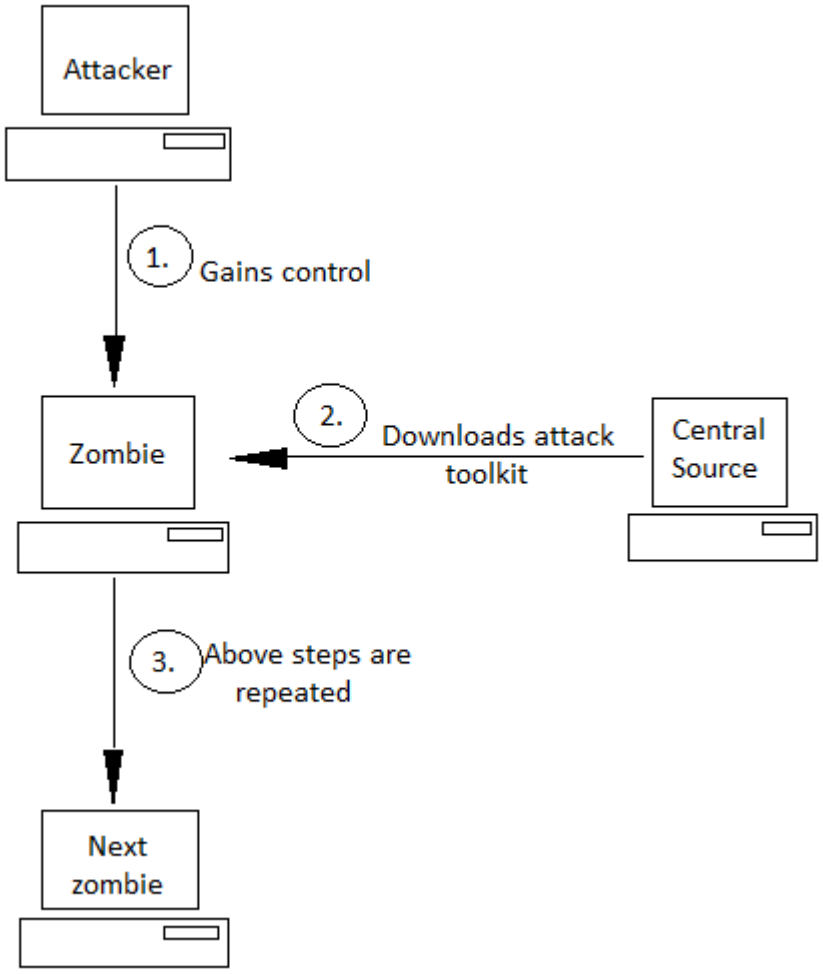

Figure 3: Central source propagation method

Back-chaining propagation

In this mechanism, the attack toolkit is transferred to the newly compromised system from the attackers' system. More specifically, the attack toolkits that are installed on the attackers' system include special methods for accepting a connection from the compromised system, and sending a file to it that contains the attack tools.

This back-channel file copy can be supported by simple port listeners that copy file contents, or by full intruder-installed Web servers, both of which use the Trivial File Transfer Protocol (TFTP).

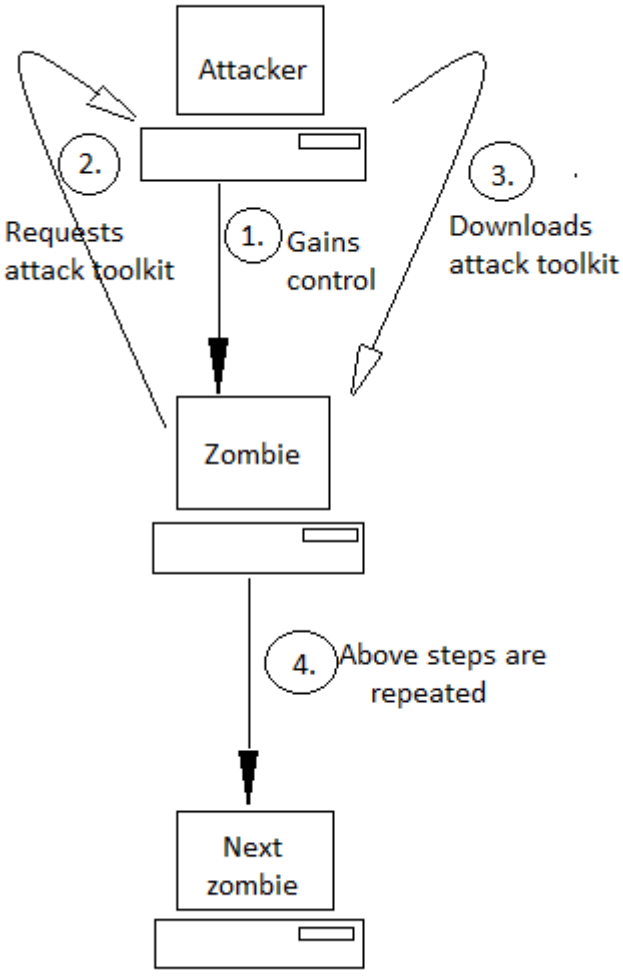

Figure 4: Back-chaining propagation

Autonomous propagation

In this method, the attackers transfer the attack toolkit to the newly compromised system at the exact moment that they break into that system. This method differs from the previously mentioned methods in that the attack tools are planted into the compromised host by the attackers themselves, and not by an external file source.

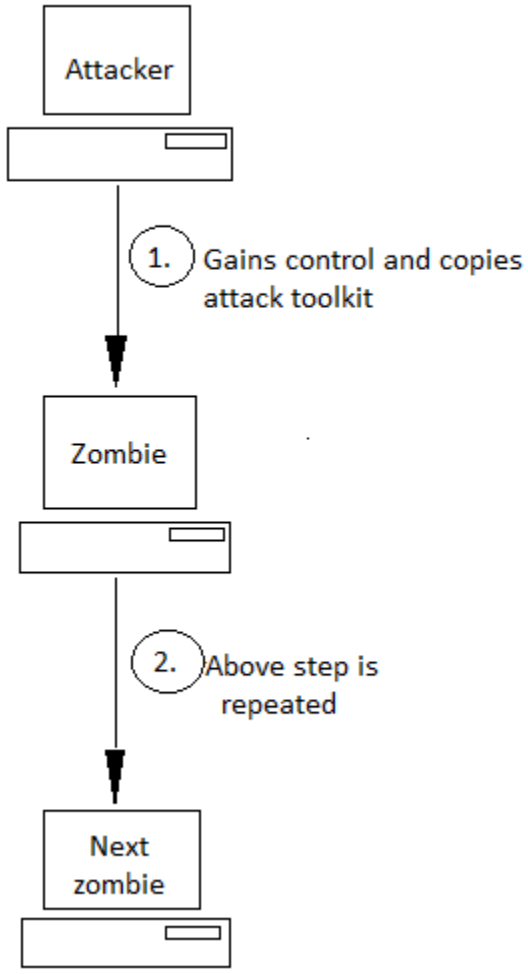

Figure 5: Autonomous propagation method

Important note: Each of these propagation mechanisms depend on the system vulnerability that's exploited, since this dictates what system rights are enjoyed by the attackers.

After the construction of the attack network, attackers use attack toolkits to specify the attack type and the victim's address, and wait for the appropriate moment in order to mount the attack. They then remotely command the launch of the chosen attack to their zombies, using the attack toolkit, to begin the attack.

The volume of traffic from a DDoS attack may be so high that the networks that connect the attacking machines to the victim may also suffer from lower

performance. Hence, the provision of services over these networks is no longer possible, and in this way, their clients are denied those services.

Attack toolkits

While there are numerous scripts that are used for scanning, compromising and infecting vulnerable machines, there are only a handful of DDoS attack tools that have been used to carry out the actual attacks.

Trinoo

This tool uses a handler/agent architecture wherein an attacker sends commands to the handler (the first system compromised in the series) via TCP, and handlers and agents communicate via UDP. Both handlers and agents are password-protected to try to prevent them from being taken over by another attacker. Trinoo generates UDP packets of a given size to random ports on one or multiple target addresses, during a specified attack interval.

Tribe Flood Network (TFN)

This tool uses a different type of handler/agent architecture. Commands are sent from the handler to all of the agents, from the command line. The attackers do not “log in” to the handler as with Trinoo. This tool can perform a UDP flood, a TCP SYN flood and Smurf attacks at specified or random victim ports. The attackers run commands from the handler using any of a number of connection methods (e.g., a remote shell bound to a TCP port, and UDP-based client/server remote shells). All commands sent from the handler to agents through ICMP packets are encoded, which hinders detection.

Tribe Flood Network 2000 (TFN2K)

An improved version of TFN, this includes several features designed specifically to make its traffic difficult to recognise and filter; to remotely execute commands; to obfuscate the true source of the traffic, and to transport TFN2K traffic over multiple transport protocols, including UDP, TCP, and ICMP. TFN2K obfuscates the true traffic source by spoofing source addresses.

Trinity

This is the first DDoS tool that is controlled via IRC. Upon compromise and infection by Trinity, each zombie joins a specified IRC channel and waits for commands. The use of a legitimate IRC service for communication between attacker and zombie replaces the classic independent handler, and elevates the level of the threat. It is also capable of launching several types of flooding attacks on a victim site, including UDP, an IP fragment, TCP SYN, TCP RST, TCP ACK, and other floods.

Now, due to regular security checks and patches, and signature-based IDS/IPS (Intrusion Detection/Prevention Systems), many of these tools have became less effective, and are not used by attackers. However, this has led to the next era of DDoS attacks, which is referred to as DDoS 2.0.

DDoS 2.0

Network firewalls today can detect the majority of flood and network DoS attacks. Many ICMP and UDP flood attacks can also be identified using intelligent packet filtering, and source and destination access-control lists. Hence, attackers today focus on application DDoS attacks, because these usually bypass most traditional network security devices.

Application DDoS attacks exploit vulnerabilities in application servers or application business logic. For example, application DDoS attacks may simply flood a Web

application server with seemingly legitimate requests designed to overwhelm Web application servers. An attacker may also attempt to exploit application vulnerabilities, such as sending Web requests with extremely long URLs.

More sophisticated attacks exploit business logic flaws. For example, if an application's website search mechanism is poorly written, it could require excessive processing by a backend database server. An application DDoS attack could exploit this vulnerability by performing thousands of search requests using wildcard search terms to overwhelm the backend application database. Moreover, the generation of session IDs, and the resources used to manage sessions, can often be overwhelmed if an attacker has the ability to generate a large number of session IDs.

Recently, "Slowloris" has emerged as a perilous application DDoS attack. It disrupts application services by exhausting Web server connections. In the Slowloris attack, the attackers send an incomplete HTTP header, and then periodically send header lines to keep the connection alive, but they never send the full header. Without requiring that much bandwidth, an attacker can open numerous connections, and overwhelm the targeted Web server. While multiple patches have been created for Apache to mitigate this vulnerability, it nonetheless demonstrates the power of more sophisticated DDoS attacks.

What makes DDoS 2.0 different?

DDoS attacks are traditionally carried out by computer-based bots. DDoS 2.0 is considered to be a highly amplified class of DDoS attacks. Recently, a new breed of DDoS attacks has been uncovered that uses Web servers as payload-carrying bots. Using a basic software program equipped with a dashboard and control panel, attackers could configure the IP, port, and duration of the attack. Hackers simply need to type the Website URL they wish to attack, and they can instantly disable targeted sites. Here are some points on why Web servers are used in DDoS 2.0:

- Servers provide a powerful DDoS attack platform, because they usually have greater bandwidth than a simple PC.
- Servers are always online, while a typical PC might go offline. Moreover, they are also rarely formatted.
- A Web server's outgoing traffic is usually less monitored by ISPs, because of a common misconception that a server's outgoing traffic is not as malicious as a PC's.
- By using Web servers as zombies, attackers are even less detectable, because tracebacks typically lead to a lone server at a random hosting company.

The tool of choice in DDoS 2.0

Do you recall the Wikileaks incident, when attacker groups supporting Wikileaks made DDoS attacks on servers of organisations that opposed Wikileaks? The tool that topped the suspected list of attack weapons was LOIC (Low-Orbit Ion Cannon).

In the current era of DDoS 2.0, LOIC is one of the first choices of attackers. It is an open source network-attack application written in C#, which performs DoS/DDoS attacks on a target site by flooding the server with TCP packets, UDP packets, or HTTP requests.

An attacker downloads the LOIC client and configures it to connect to an IRC server. The victim server gets flooded with requests from all LOIC clients, operating in "hive" mode. This is a classic Distributed Denial of Service (DDoS) using a botnet, except that in this case, attackers volunteer to join it.

If you are using this tool even for testing purposes, please be careful, because it does not include code for masking the originator's IP address, which will show up on the target server's logs and can easily be traced back to the user's ISP account, and eventually the local router. But if you want to test it on your own servers, the C# code is available at here.

In LOIC, most of the files are for creating the interface, but three of them are of interest:`frmMain.cs`, `HTTPFlooder.cs` and `Program.cs`.

The `frmMain.cs` file generates the main part of the user interface, and where the user specifies the URL or IP address of the target server, the program does a series of checks for valid addresses, port numbers, payload, etc., before running the DDoS code for whichever of the three methods (TCP, UDP or HTTP) is selected.

In the “hive” mode, commands are sent to the LOIC client through IRC. The IRC server, channel and port are set initially in the forms and defined in `Program.cs`, which uses the C# `SmartIRC4NET`library. In LOIC’s default mode, the user has volunteered to join the rest of the LOIC users all over the world, thus forming a botnet, which collectively sends mass requests to the target server.

If you face some difficulty in compiling LOIC, you can go for its binary here.

However, besides LOIC, attackers also use a variety of other tools. Some of them are availablehere.

Why DDoS attacks are difficult to detect

If a server is under a DDoS attack, it is quite difficult to detect the attack before the damage is done. The following characteristics of DDoS attacks make them very effective for attackers, and extremely challenging to defend against:

- Attack packets usually have spoofed source IP addresses. Hence, it is more difficult to trace them back to their real source. Further, it is possible that intermediate routers and ISPs may not cooperate in this attempt. Sometimes attackers, by spoofing source IP addresses, create counterfeit armies.
- The similarity of the attack traffic to legitimate traffic makes separation and filtering extremely hard. Unlike other security threats that need specially crafted packets (e.g., intrusions, worms, viruses), flooding attacks need only a high traffic volume, and can vary packet contents and header values at will.
- The easy availability of powerful DDoS tools has made it possible for even inexperienced users to bring down mighty Web servers.

Time for security

Once a DDoS attack starts on a Web server, it is extremely difficult to stop it. But, if the following security steps are practised, then the attack can be curbed to a large extent:

- Implementing CAPTCHAs can hinder automated DDoS attacks. Though bots are increasingly finding ways to circumvent CAPTCHAs, they are still an effective defense against application DDoS attacks.
- DDoS attacks are almost always performed by an automated client. Many of these client or bot agents have unique characteristics that differentiate them from regular Web browser agents. Tools/technologies that recognise bot agents, such as malware collectors and honey-pots, can help in botnet analysis, thus improving security.
- Administrators could adjust their network gateways in order to filter input and output traffic. The source IP address of output traffic should belong to the local subnet, whereas the source IP address of input traffic should not. In this way, they can reduce traffic on the network that has spoofed IP addresses.
- If you own a Web server, don’t just rely on anti-viruses, but instead use fully updated anti-malware and anti-botnet software too. In this way, you can prevent auto-installation of DDoS toolkits on your system. Moreover, always keep your systems updated and fully patched.
- Intrusion detection systems (IDS/IPS) can be a great help here in notifying the administrator if someone is trying to break in to install attack toolkits or bots.

Securing Apache from DDoS

- The limit on the number of simultaneous requests that will be served by Apache is decided by the `MaxClients` directive, and is set to 256, by default. Any connection attempts over this limit will normally be queued, up to a number based on

the ListenBacklog directive, which is 511, by default. However, it is best to increase this, to prevent TCP SYN flood attacks.

- Using traffic-shaping modules: Traffic shaping is a technique that establishes control over Web server traffic. Many Apache modules perform traffic shaping, and their goal is usually to slow down a (client) IP address, or to control the bandwidth consumption on the per-virtual-host level. On the positive side, these can also be used to prevent DDoS attacks. The following are some popular traffic shaping modules:
  - `mod_limitipconn` limits the number of simultaneous downloads permitted from a single IP address. More here.
  - `mod_throttle` is intended to reduce the load on your server, and the data transfer generated by popular virtual hosts, directories, locations, or users. Download from here.
  - `mod_bwshare` accepts or rejects HTTP requests from each client IP address, based on past downloads by that client IP address. More here.
- Apart from the above, one module that is designed specifically as a remedy for Apache DoS attacks is `mod_dosevasive` (Download link). This module will allow you to specify a maximum number of requests executed by the same IP address. If the threshold is reached, the IP address is blacklisted for the time period you specify. The only problem with this module is that users, in general, do not have unique IP addresses. Many users browse through proxies, or are hidden behind a NAT (network address translation) system. Blacklisting a proxy will cause all users behind it to be blacklisted. Hence, it is recommended to keep traffic shaping modules higher in your priority list.

Tools of the secure trade

- (D)DoS Deflate is a light-weight Bash shell script designed to assist in the process of blocking a denial of service attack.

- Apache log viewer can be used for easy analysis of Web server traffic and requests.
- Other tools such as *zmbscap-0.1* and *scrutinizer-1.03* are also effective. You can get a huge list of such tools here.

Note: I once again stress that neither I nor LFY are responsible for the misuse of the information given here. Rather, the attack techniques are meant to give you the knowledge that you need to protect your own infrastructure. Please use the tools and techniques discussed in this article, sensibly.

Source : http://www.opensourceforu.com/2011/04/securing-apache-part-8-dos-ddos-attacks/