Global Information Assurance Certification Paper

Steven H. Bass
Version 2.0 (Revised August 13, 2001)

Spoofed IP Address Distributed Denial of Service Attacks: Defense-in-Depth

One challenge to businesses worldwide is to permit and even encourage desirable Internet traffic while excluding unwanted or even detrimental traffic. Over the past 18 months, there have been numerous articles, practicals, publications, and white papers written on one particular type of unwanted and detrimental Internet traffic - Distributed Denial of Service (DDoS) attacks.

For the Internet community, the key to reducing and/or stopping DDoS attacks is to utilize a defense-in-depth approach without creating limitations in performance or scalability. The purpose of this paper is to look at a defense-in-depth approach to spoofed IP address DDoS attacks, including known defenses, new techniques, and recent developments.

Introduction

DDoS attacks against e-commerce sites demonstrate the opportunities that attackers now have because of several Internet trends and related factors: a) Attack technology is developing in an open-source environment and is evolving rapidly; b) Increasingly complex software is being written by programmers who have no training in writing secure code and are working in organizations that sacrifice the safety of their clients for speed to market; and c) User demand for new software features instead of safety, coupled with industry response to that demand, has resulted in software that is increasingly supportive of subversion, computer viruses, data theft, and other malicious acts. [1]

Ingress Filtering

DDoS attacks, which have employed forged source addresses, have proven to be a troublesome issue for Internet Service Providers. Ingress filtering is one method to reduce DDoS attacks which use forged IP addresses to be propagated from 'behind' an Internet Service Provider's (ISP) aggregation point. Ingress filtering applies to traffic received at the router from the customer. While ingress traffic filtering reduces the success of source address spoofing, it does not preclude an attacker using a forged source address of another host within the permitted prefix filter range. It does, however, ensure that when an attack of this nature does indeed occur, a network administrator can be sure that the attack is actually originating from within the known prefixes that are being advertised. This simplifies tracking down the culprit, and at worst, an administrator can block a range of source addresses until the problem is resolved. All providers of Internet connectivity are urged to implement ingress filtering to prohibit attackers from using forged source addresses that do not reside within a range of legitimately advertised prefixes. [2]

Ingress Filtering on Cisco Routers

Cisco Systems, the leading manufacturer of backbone routers, provides a Unicast Reverse Path Forwarding (RPF) feature, which helps to reduce problems caused by malformed or forged IP source addresses passing through an ingress router. When Unicast RPF is enabled on a customer interface, the router examines all packets received as input on that interface to make sure that the source address and source interface appear in the routing table and match the interface on which the packet was received. Unicast RPF checks to see if any packet received at a router interface arrives on one of the best return paths to the source of the packet. If the packet was received from one of the best reverse path routes, the packet is forwarded as normal. If there is no reverse path route on the same interface from which the packet was received, it might mean that the source address was modified or forged. If Unicast RPF does not find a reverse path for the packet, the packet is dropped. [3]

Ingress Filtering on Juniper Routers

Juniper Networks, another leading manufacturer of backbone routers, provides a similar inbound packet-filtering feature, filter-source-addr-verification, which accepts only traffic from a customer's network. Applying this filter to the interface attaching the customer will pass traffic only with a valid customer's source IP address; otherwise, it will discard traffic that has an invalid source address. [4]

Ingress Filtering Verification

The Oak Ridge National Laboratory (ORNL), sponsored by the Office of Counter Intelligence of the US Department of Energy, has developed a prototype program that an end-user can run to verify that their ISP has proper ingress filters enabled. The user can download a spoof-tester, which contacts a server with TCP and obtains a spoofed address for testing. The spoof-tester then transmits a series of spoofed packets (TCP, UDP, ICMP) from the users machine to the server. The server then notifies the spoof-tester if the spoofed packets are detected. (The actual IP address of the user's machine is embedded in the spoofed packets.) If the spoofed packets are detected, the user or testing service could then notify their ISP. (The spoofed packets transport checksums are wrong so there are no packets reflected to the spoofed address.) [5]

Traceback

While ingress filtering deals with dropping malicious packets, it is not likely to completely eliminate the ability to spoof source IP addresses because hosts within customer networks can still disguise themselves as any of the hundreds

or thousands of machines in the customer domains. As a result, techniques to traceback an attack to the source(s) are being developed. Recent proposals for traceback include a variety of packet-marking schemes for routers to use the IPv4 ID field to report information about the edges of the network that the packets traversed. The collective edge information can then be analyzed at the victim to compute the path of the attack. Weaknesses shared by all of the traceback proposals are that the damage done by an attack is not being controlled while the traceback is in progress, and the effectiveness of traceback schemes is reduced as an attack becomes more distributed. However, tracing back an attack to its source is the first step towards the necessary legal actions to discourage such attacks in the future. [6]

Intelligent Network Management/Backbone-Layer Security

Another recent development aimed at stopping DDoS attacks before they reach the customer network is an intelligent network management/backbone-layer security approach. Three companies, Asta Networks, Mazu Networks, and Arbor Networks, appear to take a similar approach, analyzing the patterns of traffic through the routers at the core and edge of the service provider networks, determining whether anomalies in the traffic suggests an attack on a router, server, or other piece of infrastructure is underway, tracing the attack back through the router system if an attack is detected, and then ultimately employing countermeasures against the attack by intelligently dropping packets or throttling back traffic over certain routers. Implementing this kind of distributed approach to detecting and pushing back DDoS attacks would require considerable coordination among the different owners of the networks and routers over which illegitimate traffic might pass. Moreover, the ability to engage in such security without significantly degrading the performance of the networks remains a question. [7]

Egress Filtering

In customer networks, it is customary to create inbound access rules to control what traffic is allowed in from the Internet. All too often however, many administrators pay little attention to what is allowed out of their network. In other words, egress filtering, or the filtering of outbound traffic is not being performed. This can make customer networks an excellent haven for DDoS attacks. The best way to ensure that only assigned IP address space leaves customer networks is to setup an outbound filter on the customers egress router. Besides ensuring that spoofing attacks cannot be launched from a customers network, it's also a great way to insure that private addressing (RFC 1918) is not leaked out. [8]

Egress Filtering on Cisco Routers

Cisco Systems, the leading manufacturer of customer premise equipment

(CPE) routers, has outlined a simple and straightforward example for applying outbound filters on CPE routers. [9]

Host-based Defense

Many host computers in user organizations are vulnerable to take-over for DDoS attacks because of inadequate implementation of well-known "best practices", such as those mentioned above. When these computers are used in attacks, the carelessness of their owners is instantly converted to major costs, headaches, and embarrassment for the owners of host computers being attacked. User organizations should check their systems periodically to determine whether they have had malicious software installed, including DDoS Trojan Horse programs. If such software is found, the system should be restored to a known good state. [1]

Host-Based Anti-Tools

One key trend is that DDoS attacks are becoming more prevalent. Additionally, DDoS attack tools are getting more sophisticated and their schemes are getting increasingly more complex. Currently, security experts have identified more than seven DDoS tools and lots of variants are appearing continuously. Knowing the enemy is the first step in stopping DDoS attacks. By having an idea about their tools and methods of attack, the Internet community can get better prepared. A few promising host-based tools that detect DDoS handlers and agents on host systems have been reviewed and are available for free on the Internet. [10]

Recent Developments with Windows XP

In 1981, The Computer Systems Research Group (CSRG), at the University of California at Berkeley, first mated the Unix operating system to the Internet. This was done by implementing Internet protocols and creating a TCP/IP Stack for Unix. The Unix operating system's built-in TCP/IP stack automatically generates and receives Internet plumbing ICMP messages. To facilitate the creation of Internet plumbing applications, such as ping and traceroute, the Berkeley designers allowed programmers to manually generate and receive their own ICMP message traffic. The Berkeley system provides this power through the use of a raw socket. A raw socket short-circuits the TCP/IP stack to open a backdoor directly into the underlying network data transport. This provides full and direct packet level Internet access to any Unix sockets programmer. Beyond their use for supporting simple ping and traceroute, the Berkeley designers intended raw sockets to be used for Internet protocol research purposes only. Because they fully appreciated the inherent danger of abuse of raw sockets, they deliberately denied raw socket access to any applications not running with maximum Unix root privileges. User-level applications were thus prevented from accessing and potentially abusing the

© SANS Institute 2000 - 2005 Author retains full rights.

raw sockets capability.

Source address spoofing requires root access on Unix systems. The attacker must have root access so that the attack software can open a raw network socket. Most applications use cooked sockets, in which the IP stack provides the necessary packet headers. A raw socket means that the application must prepare the necessary headers itself—that is, do its own cooking. This permits the attacker to put any information he or she wants in the headers, including spoofed source addresses. The most common and familiar DDoS attacks have been generated from Unix-family operating systems. Attacks launched from security-compromised Windows systems are common too. However, the Internet application-programming interface built into Windows prevented attacks from being as damaging as those launched by Unix and Linux systems. The sole reason for this difference was Windows' lack of full raw socket support. Windows Sockets (WinSock) can be readily used for their intended and safe purpose of generating valid ICMP ping and traceroute packets, and application programs are effectively cut off from direct "lower-level" access to the underlying physical Internet. As a result, traditional Windows applications were unable to spoof a machine's IP address to hide the source of any malicious traffic they might generate. However, in Windows XP, Microsoft added a number of powerful networking features because, they say, "Some people complained about Windows lack of full raw socket support". Under the Home Edition of Windows XP, all users are Administrators by default, which means the deliberately restricted raw socket interface has now become available to all system users. [11]

In response to concerns over Microsoft's decision to default all users to full administrative privilege in Windows XP, Steve Gibson and Jeremy Collake developed the SocketToMe and SocketLock programs. SocketToMe is a general-purpose raw socket availability detector. SocketToMe reveals the maximum raw socket access available to programs being executed by the logged on user. Programs may either have no raw socket access, partial (safe) access, or full (unsafe) access. Although SocketToMe can be used alone, it was created for use with the companion SocketLock program. SocketLock modifies the normal privileges of the Windows networking socket system to restrict raw socket access to the system account alone, and all background system processes continue to operate with their traditional full raw socket access, but no users accounts, not even administrators, have any access to raw sockets. Gibson and Collake created the SocketToMe and SocketLock tools to demonstrate the feasibility of implementing a simple alteration in the way Microsoft is going to expose abuse-prone raw sockets to Windows XP users. [12]

Conclusion

Defenses against DDoS attacks depend upon the Internet community working

together. With ISP's, customer network administrators, hardware and software manufacturers, and Internet security professionals implementing defense in depth best practices, together we can reduce and/or hopefully stop DDoS attacks. If you are the victim of a DDoS attack, maintaining logs of events can be very useful to understanding an attack, possibly preventing other attacks, and aiding law enforcement in finding the attacker(s).

References

[1] Consensus Roadmap for Defeating Distributed Denial of Service Attacks
A Project of the Partnership for Critical Infrastructure Security
http://www.sans.org/ddos_roadmap.htm

[2] Request for Comments: 2827 Network Ingress Filtering:
Defeating Denial of Service Attacks which employ IP Source Address Spoofing
http://www.ietf.org/rfc/rfc2827.txt

[3] Unicast Reverse Path Forwarding
http://www.cisco.com/univercd/cc/td/doc/product/software/ios111/cc111/uni_rpf.htm

[4] Minimizing the Effects of DoS Attacks
http://arachne3.juniper.net/techcenter/app_note/350001.html

[5] Backtracking Spoofed Packets
http://www.epm.ornl.gov/~dunigan/oci/bktrk.html

[6] Steven M. Bellovin – Papers
http://www.research.att.com/~smb/papers/

[7] Recent Developments and Emerging Defenses to DDoS
http://www.sans.org/infosecFAQ/DNS/developments.htm

[8] Egress Filtering v 0.2
http://www.sans.org/y2k/egress.htm

[9] Strategies to Protect Against Distributed Denial of Service (DDoS) Attacks
http://www.cisco.com/warp/public/707/newsflash.html

[10] Understanding DDOS Attack, Tools and Free Anti-tools with Recommendation
http://www.sans.org/infosecFAQ/threats/understanding_ddos.htm

[11] Why Windows XP will be the DOS Exploitation Tool of Choice
http://grc.com/dos/winxp.htm

[12] Windows XP Home Edition Must be Made More Secure
http://grc.com/dos/sockettome.htm