

Defenses against DDoS: Network, Host, & User defenses

by Gregg Zepp

A. Perimeter Defenses: One of the best places to stop a network from being a victim of a DDoS attack is at the first line of defense, the network perimeter. Devices such as firewalls, DMZ hosts, and routers can be leveraged to examine and regulate network traffic coming in and going out of an organization's network.

1. Firewalls/Router hardening.

- a. Egress Filtering. Use anti-spoof (egress) filtering to drop all outgoing traffic from the network that does not have a source IP address found within the outgoing network. These packets are representative of a mis-configured host or a spoofing attack. (DDoS attacks usually involve spoofed packets.) [1]
- b. Ingress Filtering. Block invalid inbound traffic, such as private and reserved address ranges which should normally never be honored as valid source addresses. [2]
- c. Disable directed broadcasts. As of Cisco IOS 12, directed broadcasts is enabled by default. To prevent a network from being used as an amplifying site, directed broadcast functionality should be disabled. To do this on Cisco routers: **no ip directed-broadcast**. [2] Also, consider reading RFC 2644, which states router software should deny by default forwarding and receipt of directed broadcasts. [3]
- d. Implement Unicast Reverse Path Forwarding on each interface. The router will examine all packets received to ensure the source address and source interface appear in the routing table and match the interface on which the packet was received. [2]
- e. Protect routing updates via Authentication. Do not allow unauthenticated access to the routing infrastructure. Routing protocols such as RIP and BGP have weak to no authentication. Whatever authentication can be implemented should be implemented. [2]
- f. Implement a Sink Hole. This is the configuration of a sacrificial router to advertise routes with bogus destination addresses as to establish a trap for malicious traffic. [4]
- g. Rate Limit (aka, Pushback). Cisco routers provide the rate limit command to configure Committed Access Rate (CAR) and Distributed CAR (DCAR) policies to control the amount of traffic you are willing to accept on the interface. However, if this is incorrectly implemented on interfaces, some networks may lose too much traffic and degrade performance. [2]

2. **Network Intrusion Detection Systems (NIDS).** *Network based intrusion detection attempts to identify unauthorized, illicit, and anomalous behavior based solely on network traffic. A network IDS, using either a network tap, span port, or hub collects packets that traverse a given network. Using the captured data, the IDS system processes and flags any suspicious traffic. Unlike an intrusion prevention system, an intrusion detection system does not actively block network traffic. The role of a network IDS is passive, only gathering, identifying, logging and alerting.* [5] Companies who make NIDS appliances are: Cisco, TippingPoint, and Blue Coat.

3. **Cisco specific hardening.** There are other vulnerabilities and countermeasures specific to Cisco devices. The book, *Hacking Exposed: Cisco Networks*, has a chapter devoted to Cisco DoS attacks and mitigations. [6]

B. Host Defenses: One of the primary concerns here is to keep the system from getting any kind of malware, Trojan, rootkit, or bot on in for then it can be used as a "zombie" to be part of a network of systems who are

used together for malicious activity, such as launching a DDoS attack.

1. **Keep systems patched.** Apply hot fixes, patches, security updates in a timely manner.
2. **Host Intrusion Detection System (HIDS).** *A host-based IDS monitors all or parts of the dynamic behavior and the state of a computer system. Much as a NIDS will dynamically inspect network packets, a HIDS might detect which program accesses what resources and discover that, for example, a word-processor has suddenly and inexplicably started modifying the system password database. Similarly a HIDS might look at the state of a system, its stored information, whether in RAM, in the file system, log files or elsewhere; and check that the contents of these appear as expected.* [7]
3. **Disable unnecessary services.** The more services running on a host, the more ports are listening to connect via those services to other network devices and hosts. Many services have programming flaws which allow a DoS to occur from as little as one malformed packet. [9] Limit open ports and connectivity opportunities to networked hosts by disabling unnecessary OS services from starting up. This can be done a number of different ways within each different OS. See Google Search: ["disable unnecessary services" security linux windows](#)
4. **Anti-virus/Anti-Malware/Anti-Rootkit software.** There are a number of commercial and even freeware antivirus and anti-rootkits programs available. What is a "rootkit"? *A rootkit is a software or hardware device designed to gain administrator-level control over a computer system without being detected. Although rootkits can serve a variety of ends, they gained notoriety as malware, appropriating computing resources without the knowledge of the administrators or users of affected systems. Rootkits can target the BIOS, hypervisor, boot loader, kernel or less commonly, libraries or applications.* [8] A rootkit is often used to implement a bot on a host eventually turning it into a zombie machine part of a larger botnet network. (Please see this week's forum post by Robert Leung for "What are botnets.")
5. **File integrity checkers.** *It is very difficult to compromise a system without altering a system file, so file integrity checkers are an important capability in intrusion detection. A file integrity checker computes a checksum for every guarded file and stores this. At a later time you can compute a checksum again and test the current value against the stored value to determine if the file has been modified.* [10] For a list of well-known integrity checkers: www.networkintrusion.co.uk/index.php/products/IDS-and-IPS/File-Integrity-Checkers.html
- C. **User Defenses:** Locking down high-use applications that allow users to access resources from other networks, such as an internet web browser [9], can help limit malicious content from getting on a workstation. Also, training system users what to look for regarding malicious content and having a network usage policy can help protect the network by having a security aware systems user base.
1. **Don't let Non-Admin Users log in as Administrators.** Give personnel only those authentication and permission configurations needed to perform their jobs. Giving too much permission to a non-savvy system user could be disastrous for the company's network. It also allows for malicious activity by savvy system user that would not be in the organization's best interest. It's commonly said that 80% of network attacks are performed by those who are, or at one time were, users (insiders) of the same network.

2. **Exercise safe internet usage.** Although there are a number of browser settings which can help restrict browser activity by a user, training users to spot potentially malicious sites, files, etc, can be another line of defense for the network.
3. **Exercise safe email usage.** Inform system users to try not to open email from addresses they do not recognize. Especially, do not open untrusted email attachments. Also, set AV/AM programs to scan email messages and attachments.

Citations -----

1. Ed Skoudis, *Counter Hack Reloaded*, (Prentice Hall, 2006); 542-543.
2. Stuart McClure; Joel Scambrary; George Kurtz, *Hacking Exposed 6*, (McGraw Hill, 2009); 652-653.
3. RFC 2644: Changing the Default for Directed Broadcasts in Route (www.faqs.org/rfcs/rfc2644.html)
4. Arbor Networks; "Sink Holes" presentation. (www.arbornetworks.com/dmdocuments/Sinkhole_Tutorial_June03.pdf)
5. SANS.org: www.sans.org/security-resources/idfaq/what_is_id.php
6. Dr. Andrew Vladimirov et all, *Hacking Exposed: Cisco Networks*, (McGraw-Hill, 2006); 369-375.
7. Wikipedia: en.wikipedia.org/wiki/Host-based_intrusion_detection_system
8. Wikipedia: en.wikipedia.org/wiki/Rootkit
9. Roger Grimes, *Windows Desktop and Server Hardening*, (Wiley Publishing, 2006); 254.
10. SANS.org: www.sans.org/security-resources/idfaq/integrity_checker.php