

# Understanding TCP IP to Prevent Network Attacks - Part 2

Michael J. Martin

In part one of this series, I began a discussion on network exploits and some of the available IOS options for defending your Internet connection. The first part covered TCP/IP operation and this article will follow up with vulnerabilities, attacks and how to defend the homeland.

## Vulnerabilities

While TCP/IP is great for delivering packets in meshed networks, it is built upon network transmission facilities of questionable reliability. It pays a price in terms of security for that versatility. The following reviews a number of TCP/IP exploits that administrators should be aware of and some actions that can be taken to defend against them.

## IP Spoofing

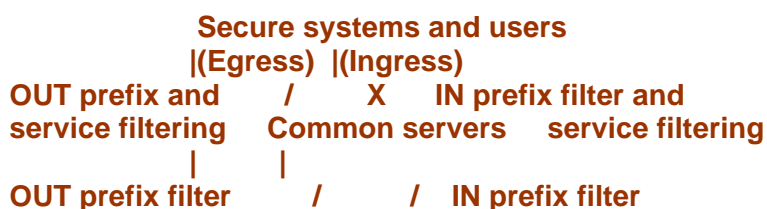
In IP spoofing, a trusted host is compromised or disabled by an attacker. The attacker then sends IP packets with the source address of the disabled trusted host to the target, effectively masquerading as the trusted host to gain access to or disable the target host. Technically, IP spoofing is not really an attack. It is a vulnerability that is exploited in order to execute an attack. The idea of granting access privileges based on IP address dates back to the TCP/IP's origins. This practice is quite prevalent in the IP world as a means for enforcing system security. Static router access control lists (ACLs) and the Unix tool TCP Wrappers both work on this principle. The most advanced attack utilizing this exploit (in conjunction with TCP sequence number prediction, which will be discussed ahead) is session hijacking, with Hunt and Juggernaut being the most "visible" of the number tools available today. Simpler, but no less effective attacks involve sending "crafted" packets to disrupt open sessions and directed and distributed denial of service (DDoS) attacks.

Network administrators can take several preventative measures to protect against IP spoofing exploits. No trusted services: Do not use network services that only rely on address-based authentication. Address-based authentication is fine as long as it is used in conjunction with a unique credential-based method, such as SSH, kerberised telnet or r\* commands, or a one-time password (OTP) authentication system (i.e., S/Key or Security Dynamics' SECUREID).

External access filtering: RFC 2827 recommends the ingress and egress filtering of RFC 1918 address and local prefix addresses. Filtering RFC 1918 addresses prevents hosts in "your" network from launching DoS and DDoS attacks using unroutable/unreachable addresses. Filtering traffic with source address matching your local prefix prevents outside-originated IP-spoofing-dependent attacks.

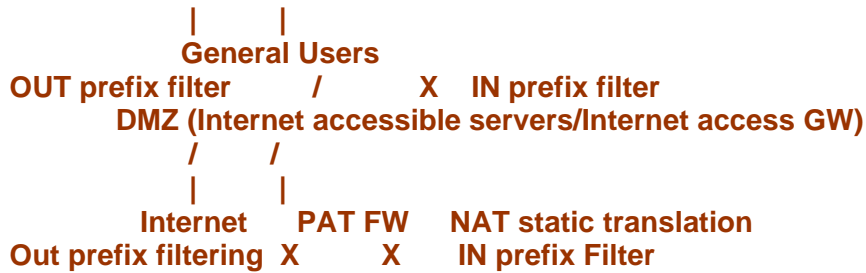
Internal access filtering: The majority of IP-spoofing-based attacks are launched over the LAN. ACLs can be used to minimize risk by imposing an "explicit-permit" access model where communication exchange between network segments, users, and servers all need to be defined. The easiest (and the most costly) thing you can do to secure your LAN is move to an all-switched network, permitting only one host per port. Shared segment networks allow users to see each other's data transmissions, making "sniffing" easy. Permitting more than one host on a switch port creates two potential problems: 1. Users on the hub can sniff each other's transactions. 2. Hubs are often installed in accessible places, making it easy for someone to "plug-in" unnoticed.

Ideally, the LAN's should be architected using a layered approach.



# Understanding TCP IP to Prevent Network Attacks - Part 2

Michael J. Martin



The "inside core" contains the most secure systems (i.e., financial systems, human resources, research and development, etc.) and their users are on the same subnet with inbound and outbound static filters. If your Ethernet switch supports media access control (MAC) filtering or private VLAN support, user-to-server access control within the local IP subnet can be implemented to provide an additional layer of security. General user segments should be on separate subnets from general server segments. Access control filtering only needs to explicitly permit the IP subnets in use within the administrative domain. Random dynamic IP address assignment for users (basic DHCP) is a major no-no.

In a perfect environment, IP addresses should be statically assigned to hosts that remain connected to the network (no laptops), especially in high-security networks. With the prevalence of laptops and today's "expanded workday," a user environment comprised only of static hosts is not practical. The next-best thing is a dynamic user environment where static IP addresses are assigned using DHCP reservations. The goal in constructing a secure LAN environment is auditability. Network and system administrators should be able to monitor user activity and server access with moderate ease. The easiest way to facilitate this is using static IP address assignments and access control filtering that logs illegal activity by explicitly permitting service access.

## Smurf Attacks

A smurf attack is a type of denial of service attack that exploits the use of the Internet Control Message Protocol (ICMP) and IP prefixes network and broadcast addresses. A smurf attack's purpose is to disable a target host or network by consuming all of its bandwidth; aside from this it causes no permanent damage. Every IP prefix has two special addresses: the network address which is the first address in the prefix, and the network broadcast address that is the last address in the prefix range. The IP network address serves as the identity address of a given prefix in an IP routing table. The IP broadcast address was devised as a method for sending information to all of the nodes in the prefix. Most IP implementations will respond to messages with the network or broadcast address as the source address. This support is known as "directed broadcast." This feature is also rarely used for legitimate purposes. For example, here is a cheap trick to estimate how many hosts are active on a prefix: Run the ping command using the -c flag sending two packets, with a packet size (using the -s flag) of 1 byte (9 bytes total size including the ICMP header) with the IP prefix broadcast address as the target address. Any host that supports the processing of "directed broadcast" messages will respond. Here is a command line example from a BSD (Berkeley Software Distribution) system:

```
bash2.05 mmartin@trinity ~ % /sbin/ping -c 2 -s 1 172.30.170.255
PING 172.30.170.255 (172.30.170.255): 1 data bytes
9 bytes from 172.30.170.254: icmp_seq=0 ttl=255
9 bytes from 172.30.170.8: icmp_seq=0 ttl=255 (DUP!)
9 bytes from 172.30.170.5: icmp_seq=0 ttl=255 (DUP!)
9 bytes from 172.30.170.64: icmp_seq=0 ttl=255 (DUP!)
9 bytes from 172.30.170.3: icmp_seq=0 ttl=255 (DUP!)
```

# Understanding TCP IP to Prevent Network Attacks - Part 2

Michael J. Martin

```
9 bytes from 172.30.170.4: icmp_seq=0 ttl=255 (DUP!)
9 bytes from 172.30.170.254: icmp_seq=1 ttl=255
```

```
--- 172.30.170.255 ping statistics ---
2 packets transmitted, 2 packets received, +5 duplicates, 0% packet loss
bash2.05 mmartin@trinity ~ %
```

In the example above, you should notice two things. First, of the six hosts responding, five them have a (DUP!) marker after their response. The second one does not. The host without the (DUP!) marker is the first host the source host received a response from; the remaining hosts are interpreted as duplicate replies. This condition can also occur if two hosts are assigned the same IP address or if a host's subnet mask is set incorrectly. The first responding host will have the fastest response time and will vary if the test is run over a period of time, depending on the load on the network and respective hosts and their physical distance from the source host. And in case you're wondering why we send such a small ICMP packet, this test can place a significant load on the network, depending on the number of hosts and size of the message. It is the number of hosts and the traffic load they can potentially generate that the smurf attack exploits.

A smurf attack works by sending an ICMP echo request to a prefix's broadcast address with a spoofed IP address. The result is that the actual host with the spoofed address is bombarded with hundreds and possibly thousands of ICMP echo replies. This attack is often launched using multiple IP networks, sending echo-reply messages to a single host. It not only effectively disables the target, it also consumes all of the network bandwidth of the drone networks being used to launch the attack. In order to use a network to launch a smurf attack, the network has to support the forwarding of packets with the prefix's broadcast address. Networks that allow the forwarding of directed broadcast packets are known as smurf amplifiers.

Smurf amplifiers also make it possible to launch smurf DoS attacks. Most operating systems and routers provide the ability to disable directed broadcast forwarding. Cisco routers running IOS code versions prior to 12.0 can disable directed broadcast support using the interface configuration command `<no ip directed-broadcast>`. This must be configured on each active router interface. In IOS version 12.0 and above, directed broadcast support is disabled by default. Additional, protection for Internet accessible networks can be gained by disabling directed broadcast support on hosts as well. These actions prevent your site from being used in an attack.

Two Web sites, <http://www.powertech.no/smurf/> and <http://www.netscan.org/>, scan the Internet for networks that support directed broadcast forwarding. Both list offending networks on their sites, giving network administrators the information they need to filter out inbound traffic from these prefixes. While this sounds harsh, filtering it is really the only way to protect your location from possible attack. But before you filter anything, check in with your security team and discuss the possible impact of this defense in terms of lost access and then decide if it is warranted.

## Source Routing

IP source routing (Option ID 137, strict source routing [SSR] and Option ID 131, loose source routing [LSR]) is an IP forwarding option. Packets sent with this option record the hop path (up to nine ISs) in the options data field. The most common implementation is the ping command with the `-r` flag and traceroute with the `-g` flag. These options were created for troubleshooting conditions where multiple paths exist. While valuable for problem solving, IP source routing presents some unique security problems. The "classic" attack is where an attacker sends requests using packets with the spoofed address of a trusted host with the source-route option enabled. The attacker monitors the "path," intercepts the packets and replies accordingly, or disables the trusted host and forces the traffic to

# Understanding TCP IP to Prevent Network Attacks - Part 2

Michael J. Martin

another host that has assumed its identity, using a specific routing path. The compromised host is unaware of the attack, since it assumes it is communicating with a legitimate host.

A more recent use for IP source routing is in monitoring an attack. An attack is launched using a spoofed source address and the source routing option. The attacker monitors a legitimate IS or directs the traffic flow through a controlled system allowing the attacker to monitor undetected. The majority of Unix implementations and unpatched Windows variations will accept packets with the source routing option. Despite its diagnostic advantages, it is a common best practice to disable support for source routing on ISSs. To disable source routing on Cisco routers running IOS, the global configuration command `<no ip source-route>` is used. When enabled, IP packets with the SSR or LSR IP option flag will be discarded.

## Routing table Corruption

The idea of routing table corruption was first mentioned in S.M. Bellovin's "Security Problems in the TCP/IP Protocol Suite," which was published about a year after the Morris Internet worm attack. In this paper, Bellovin discusses the potential problem of having routing information accepted and acted upon unchecked. The attack forces all or at least some (a specific host's) network traffic to a specific host using nothing more than the gateways routing table. Bellovin called for the implementation of authentication and verification in both interior and exterior gateway routing protocols (IGRPs and EGRPs) i.e., RIP, OSPF and BGP. When the article was published in 1989, such support was not available. Today, RIPv2, OSPF, and the EGRP Border Gateway Protocol (BGP) all support peer-based authentication. And additional routing table control can be added using IOS `<distribution lists>` to filter-routing announcements. Unfortunately, network engineers generally ignore this easily fixable, well-known vulnerability. Take a look at your routing table; are there routes that should not be there, or do you have peers you know nothing about?

## Denial of Service Attacks

While ICMP smurf attacks are perhaps the most well known, there are a number of TCP and UDP-based DoS attacks. DoS attacks are about blocking access, not damaging data or systems, like viruses. DoS attacks come in two flavors: host-to-host or just plain. DoS and DDoS attacks use a many-to-one attack paradigm. To start, we will look at host-to-host DoS attacks.

Land attack (impossible IP attack): This is an IP spoof-based attack where the source and destination address are the same. This attack crashes some TCP/IP implementations that do not know how to handle the packet. This is a rarity in terms of appearance in the real world, but is a standard signature on ISS, NFR, Dragon and Cisco Net Ranger and IDS-IOS.

Xmas tree attack: A TCP packet sent to any known service port will set all of the code flags (URG, ACK, PSH, RST, SYN, FIN). An alternative version of this attack is a TCP packet with no flags set. Both cases are the result of packet craft and will not exist in the "wild."

Teardrop: This attack uses fragmented UDP packets. The first fragment is fine, but the second packet overwrites part of the first fragmented packet. This results in a memory error and the system crashes.

TCP/UDP diag services attacks: There are two variations on this attack. The first is a flood of spoofed packets to the echo service (UDP/TCPport 7), which is a simple service that echoes back any data sent to it. The other involves sending a spoofed UDP message appearing to be from the chargen service port to the echo (UDP port7) service on another system. The chargen service responds to any packet sent to the service port with a 72-byte random character string. Once the spoofed connection is established, the echo port sends traffic to the chargen port and a loop develops. Both variations consume CPU resources; enough attacks will cause the system to become CPU-bound and crash.

# Understanding TCP IP to Prevent Network Attacks - Part 2

Michael J. Martin

**Ping of death:** This attack, like teardrop, exploits IP's fragmentation capability. The attack host sends an ICMP with a packet size that exceeds the maximum IP datagram size of 65,535 bytes. The attacked system waits until all of the fragments are delivered, and then reassembles the packet, resulting in a buffer overflow that crashes the system.

**SYN flood (half-open attack):** Each TCP open (SYN) request requires a server to reserve resources to support the connection. If you want to check the state of the IP connections on a Unix and NT system, you use the `<netstat -na>` command. The attack works by flooding a service with SYN requests from spoofed IP addresses (routable or unroutable addresses will work). The server then acknowledges the SYN requests and responds with SYN-ACKs. These are never responded to, because sources of the SYN requests never made the request. Depending on the system, the connection timer can allow between 1 to 3 minutes before reclaiming the allocated resources. The actual expected response to SYN-ACK is a few milliseconds. So with a steady stream of SYN requests, the server's connection queue can be filled with incomplete connections blocking out legitimate user requests.

The impact of this attack varies depending on the service attacked, but overall only server access is affected. Once the attack stops, the service will be restored; systems typically will not crash from a SYN flood attack. SYN attacks are a very common false alarm for intrusion detection systems. Unless the SYN attack is executed using an unroutable source address (which you should be filtering), the difference between a SYN attack and troublesome connection attempt can be hard to discern. SYN attacks can also be used for surveillance for session hijacking.

**Session hijacking:** This type of attack is not about DoS, but about taking over an established session to either gain access to the system, crash the system by inserting a buffer overflow attack, or simply terminating open sessions using RST packets.

There are a number of DDoS attacks that are both manmade and the result of OS bugs. All, however, work from the same basic premise. They inundate a target system with more data than it can possibly process over its network connection. There have been two basic trends in the development of DDoS attacks. The first DDoS attacks started appearing in 1998 and 1999 as viral Trojan horses (i.e., AntiBTC, BTC, and FBTC attacks). The virus trend continues today, with Word and Excel VB macro viruses like Melissa and Papa, and viruses like code-red. The other trend has been the development of commander and fleet (C&F) attacks.

The C&F DDoS attacks follow "spread the burden" attack pathology. A DDoS attack has two components: a master client/handler (which can be two different components), and agents or "zombies." The most important thing to the attacker is to avoid being caught. So the attack console is used to communicate to the handler applications, which in turn control the attack agents that actually perform the attack.

In terms of actual impact, both the attack target and attack perpetrators (the zombies) are the victims of DDoS attacks. The attack targets are the loss of service of the attacked host and the degraded performance of the network. The penetrations of "agent" systems represent a more significant security issue, because the successful use of DDoS tools requires the penetration and installation of handlers and agent software on hundreds and even thousands of systems undetected by their system administrators. Here is an overview of the most common DDoS tools.

**Trinno:** This was the "first" C&F DDoS tool, used to attack the University of Minnesota. Trinno works by flooding the target with UDP traffic. It uses TCP ports 1524 and 27665, and UDP ports 27444 and 31355 to send command and control (CC) data between master, handlers and agents. The attack

# Understanding TCP IP to Prevent Network Attacks - Part 2

Michael J. Martin

does not use spoofed IP addresses and the CC ports are defaults that can be changed by the attack implementer.

Tribe flood network (TFN) and tribe flood network 2k (TFN2K): These pick up where Trinno falls short. TFN support attacks support both spoofed and non-spoofed attacks using UDP, ICMP, and TCP SYN floods. All CC messaging is done using ICMP echo and ICMP echo-reply messages, making it hard to defend against using SACLS. TFN2K added an encrypted one-way CC method, making it almost impossible to trace back to the handler. Complete analysis of TFN is available at Washington University.

Stacheldraht: The German word for "barbed wire," stacheldraht is based on the TFN code tree. Master client-to-handler CC uses TCP port 16660. Handler-to-agent communication uses TCP port 65000 and ICMP echo-reply for CC messaging. And like TFN2K, all communication is encrypted. Complete analysis of stacheldraht is also available at Washington University.

The fact that these tools exist and have been successfully used clearly proves the need for more education on securing systems for Internet use and the more effective use of intrusion detection systems.

I hope these overviews of protocol operation and vulnerabilities are helpful to those of you getting started with intrusion detection. This is by no means a complete list of potential security vulnerabilities; it is intended to be a starting point only. There are a number of excellent books on the subject along with numerous online resources I would encourage you to examine. Next month we will look at creating Cisco IOS-named SACLS for defending your Internet gateway, so you can better defend your network homeland.