Interested in learning more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

# ICMP Attacks Illustrated

The simplicity of the ICMP protocol and the lack of awareness of security issues related to protocol has led me to put in place this paper to attempt to illustrate some of the possible attacks using ICMP as a tool.

Copyright SANS Institute
Author Retains Full Rights

AD
Symantec™
Endpoint Protection 12
The next generation of reputation-based security
Download the beta
Symantec.
Confidence in a connected world.

## ICMP Attacks Illustrated

The simplicity of the ICMP protocol and the lack of awareness of security issues related to protocol has led me to put in place this paper to attempt to illustrate some of the possible attacks using ICMP as a tool.

Also included in this paper are references to some of the tools that are available for use and in some instances, these have been used for some real world attacks.

## ICMP Basics

ICMP, the Internet Control Message Protocol is an integral part of any IP implementation.

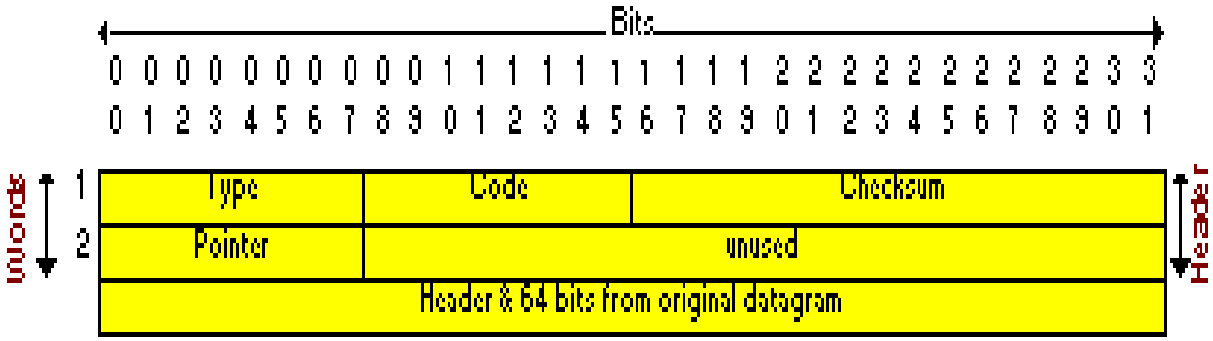
Although ICMP messages are sent in IP packets and it uses IP as if it were a higher-level protocol, ICMP is in fact an internal part of IP, and must be implemented in every IP module. ICMP messages are classified into 2 main categories:

- ICMP Error Messages
- ICMP Query Messages

Its goals and features as outlined in RFC 792 is to provide a means to send error messages for non-transient error conditions, and to provide a way to probe the network in order to determine general characteristics about the network.

A number code, also known as the "message type", is assigned to each ICMP message; it specifies the type of the message. Another number code represents a "code" for the specified ICMP type; it acts as a sub-type, and its interpretation is dependent upon the message type.

The diagram below shows the general ICMP packet format.




© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

**Attacks Illustrated**

Phase I – Reconnaissance & Scanning

ICMP Sweep

In any typical attack scenario, the attacker will first engage in some reconnaissance and scanning activities in order to

1. Better understand the environment of the target
2. Gather information about the target so as to plan the attack approach
3. Employ the right techniques & tools for the subsequent attack phases

One of the most common (albeit noisy) and most well understood technique for discovering the range of hosts which are alive in the target's environment is to perform a **ICMP sweep** of the entire target's network range.

An ICMP sweep involves essentially sending a series of ICMP request packets to the target network range and from the list of ICMP replies infer whether certain hosts are alive and connected to the target's network for further probing.

Although the above attack can be done manually via a very simple command **ping**, many automated scanning tools (E.g. nmap (http://www.insecure.org/nmap) and Superscan (http://www.foundstone.com/rdlabs/proddesc/superscan.html)) will speed up the entire scanning process by performing such a scan on all possible IP address range given a target network.

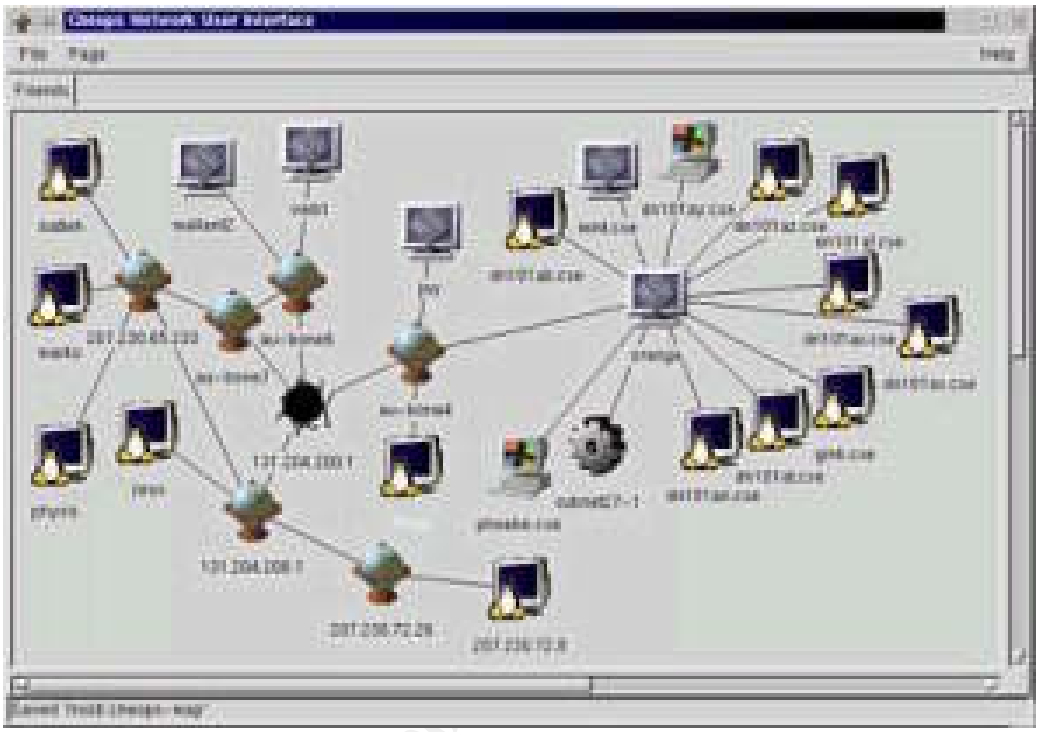
Traceroute

Another very useful tool for mapping out the target's network configuration is the use of a very simple command call **traceroute**. What this command essentially does is, it will send out progressively a series of packets with an increasing TTL (Time to Live) value set. When an intermediate router receives a forwarding packet, it'll decrement the TTL value of the packet before forwarding it to the next router. At this time if the TTL value of the packet reaches zero, an ICMP "time exceeded" message will be send back to the originating host. By sending the packet with initial TTL value of 1 will allow the first router in the path of the packet to now send back an ICMP "time exceeded" message which will then allow the attacker to know the IP address of the first router. Subsequent packets are send by increasing the TTL value in the packet by 1 each time, thus the attacker will be able to know every hop between him and the target.

Using this technique, the attacker could not only trace the path taken by a packet as it travels to the target but also gives him information on the topology of the target network. This information is crucial in allowing the attacker plan his approach when attacking the network.

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

A network-mapping tool like Cheops (http://www.marko.net/cheops) would allow the attacker to quickly map out the entire target network using ping and traceroute. This tool is a very noisy tool from a traffic perspective and can be easily picked up by any intrusion detection system as well as firewall logs.



Firewalk

Developing further from the traceroute idea, this next technique (Firewalk) can be used to identify ports that are open on a packet filtering firewall. The purpose of doing so is really to map out the filtering rules that are being set up in a packet filtering firewall.

Firewalking is typically done in 2 phases, phase 1 involves doing a traceroute from the attacker to the target firewall to ascertain the number of hops it will take for a packet to reach the firewall. During the scanning phase, TTL value of packets will be set to one greater than the firewall and send to a known host behind the firewall. If an ICMP "time exceeded" message is received, that would mean that the packet has managed to get past the firewall and thus causing an ICMP packet to be returned by the known host because TTL value has now reached zero, otherwise it can be deduced that there is a filtering rule on the firewall which stops the traffic.

Firewalk can be found at (http://www.packetstormsecurity.com/UNIX/audit/firewalk/).

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

Inverse Mapping

Inverse Mapping is a technique used to map internal networks or hosts that are protected by a filtering device. Usually some of those systems are not reachable from the Internet. We use routers, which will give away internal architecture information of a network, even if the question they were asked does not make any sense, for this scanning type. We compile a list of IP's that list what is not there, and use it to conclude where things probably are.

An Inverse Mapping attack is illustrated below:

Step 1. Attacker sends an ICMP reply message to a range of IP addresses presumably behind a filtering device.

Step 2. Upon receiving the series of ICMP reply messages, since the filtering device does not keep state of the list of ICMP requests, it will allow these packets to their destination.

Step 3. If there is an internal router, the router will respond with a ICMP "Host Unreachable" for every host that it cannot reach, thus giving the attacker knowledge of all hosts which are present behind the filtering device.

OS Fingerprinting

Before any attack can be launched, other than knowing the existence of the target host, it would be extremely beneficial to know the underlying operating system as well as the list of services that it runs. While port scanners can determine the types of services that are being offered on the system, ICMP could again be engaged in helping the attacker determine the underlying operating system.

The advantage of using ICMP protocol in a remote OS fingerprinting exercise offers the attacker a more stealthy way in OS identification process. In some instances only a single packet is sent to determine the operating system used by the target system.

Remote OS Fingerprinting is a technique that exploits the fact that different operating system vendors have built a slightly different way of handling network traffic. A detailed study of both active and passive remote OS fingerprinting was done and a detailed report can be found at (http://www.sys-security.com/html/projects/X.html).

A remote OS Fingerprinting attack is illustrated below:

Step 1. Attacker sends an UDP packet with DF bit set to a target host whose UDP port is closed.

Step 2. An ICMP "Destination unreachable port" message will be returned to the attacker.

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

Step 3. Due to the fact that different hosts will send a slightly different ICMP packet back, operating systems can be determined by examining several bits in the return packet.

E.g. If we look at the precedence bits field of the packet and the value is 0xc0, the underlying operating system can most likely be deduced to be a Linux kernel 2.0.x / 2.2.x /2.4.x based machine or a Cisco based router or a Extreme Networks switch.

In this instance, to differentiate between the Linux kernel and that of the networking device, ICMP Error Quoting size fingerprinting method can be employed. In this method, the returned ICMP packet is inspected for the number of bytes that are being returned. Linux kernel will return a different number of bytes as compared to networking device, thus we are able to differentiate them.

One step further is to be able to differentiate between the various versions of the Linux kernel. In this case we will be looking at the IP TTL value set in the packet, Linux kernel 2.0.x has got an initial value of 64 whereas 2.2.x and 2.4.x will use an initial value of 255. Now to differentiate between the 2.2.x and 2.4.x is to look at the IP ID value of the packet, 2.4.1 – 2.4.4 has got a value equals to zero unlike 2.2.x. Thus just by looking at 1 return packet from the target, the attacker is able to drill down to the type and version of the underlying operating system.

Other techniques like checking how a host responds to a crafted ICMP timestamp request are also employed to differentiate between operating systems.

X Probe available at (http://xprobe.sourceforge.net/) is a tool build to allow you do the above in an automated manner.

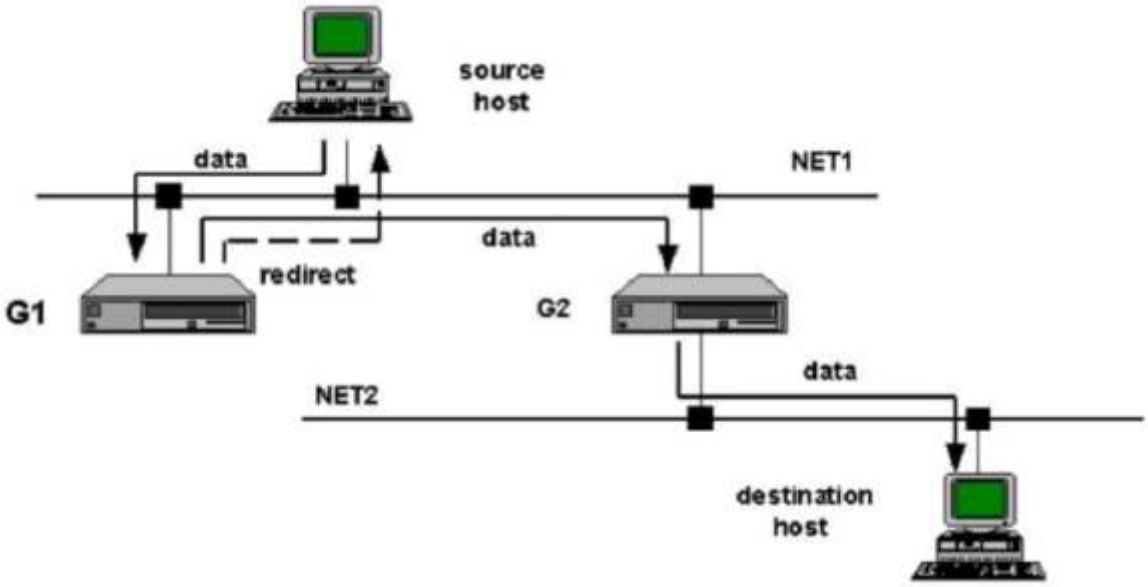
Phase 2 – Exploiting Systems

ICMP Route Redirect

An ICMP Route Redirect message is sent when a gateway receives an IP traffic from a host and finds in its routing table that its next gateway to be routed to for this traffic is on the same network as the host.

A first look at this does not really reveal any problems with this, but let's go through a scenario to see how this could be exploited to allow a Man-In-The-Middle attack to be launched.

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.




Step 1. Attacker manages to take over a secondary gateway G1 of the source host.

Step 2. Attacker sends a TCP open packet to source host acting as destination host.

Step 3. While a reply is in transit from the source host to the destination host through gateway G2, the attacker sends an ICMP route redirect message to source host spoofing as G2.

Step 4. Source host will accept the route change control message as valid and thus changes its routing table to now route all traffic bound for destination host through Gateway G1.

Step 5. Now attacker will quietly read/modify and forward all traffic bound for destination host to Gateway G2 acting as a Man-In-The-Middle host.

ICMP informational messages

By sending "oversized" ICMP messages to a target host could potentially crash / reboot the target host. This is due to the fact that some OS does not know how to handle packets that are larger than the maximum size as stipulated in RFC.

The TCP/IP specification allows for a maximum of 65536 octets in a single packet of information. This exploit can easily be exploited through the use of the ping command (with a flag to indicate the size of the packet to be send) by using a packet size greater than 65536 octets. Some OS will perform checks on the size of the outgoing ping packets and will not allow packets greater than 65536 octets. There are many tools that are available for download that will allow the attacker to create customized ping packets. One such example is hping2 (http://www.securityfocus.com/tools/641).

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

If the target host is not properly patched, the OS will freeze or reboot after receiving just 1 oversized packet.

By exploiting the nature of fragmentation as well as oversized ICMP packets, another exploit is possible that will cause some OS to stop responding and have to resort to a reboot to recover from these attacks.

SSPing (http://packetstormsecurity.org/Exploit_Code_Archive/ssping.zip) is a tool that does just that.

Developing further from this idea is another tool Jolt2 (http://razor.bindview.com/publish/advisories/adv_Jolt2.html)

In this attack, sending large numbers of identical fragmented IP packets to the target host will cause the host to stop responding for the period of time when the attack is in progress.

Another tool teardrop (http://packetstormsecurity.org/Exploit_Code_Archive/teardrop.c) sends a stream of fragmented packets to the target host and asks it to put them back together. When the host tries to do so, it discovers that the packets are not the size they say they are. This causes the target host to hang and require a re-boot before it will function again.

ICMP Router Discovery Messages

Before a host is able to send a message to a host outside its own subnet, it must be able to identify the address of the immediate router. This is typically done through reading a configuration file upon startup and on some multicast network by listening to routing protocol traffic.

An extension to the ICMP protocol called "ICMP Router Discovery Protocol" (defined in RFC 1256 - http://www.faqs.org/rfcs/rfc1256.html) is able to use "router advertisement" as well as "router solicitation" messages to allow hosts to find out the IP addresses of the router that is attached to their immediate network.

When a host is being started up, it will make use of the "router solicitation" messages to check for the address of the immediate router. Since these messages are not authenticated, attackers on the same subnet as the host can spoof these messages.

A possible attack scenario is illustrated below:

Step 1. Host boots up and issues a "router solicitation" message to find out the default router on the network.

Step 2. Attacker listens in to the message and spoofs a reply to that host.

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

Step 3. The default route of the host is now set to the attacker's IP address that the attacker has included in his reply.

Step 4. Now the attacker could employ either sniffing, man-in-the-middle attack for all traffic outbound through the attacker's machine.

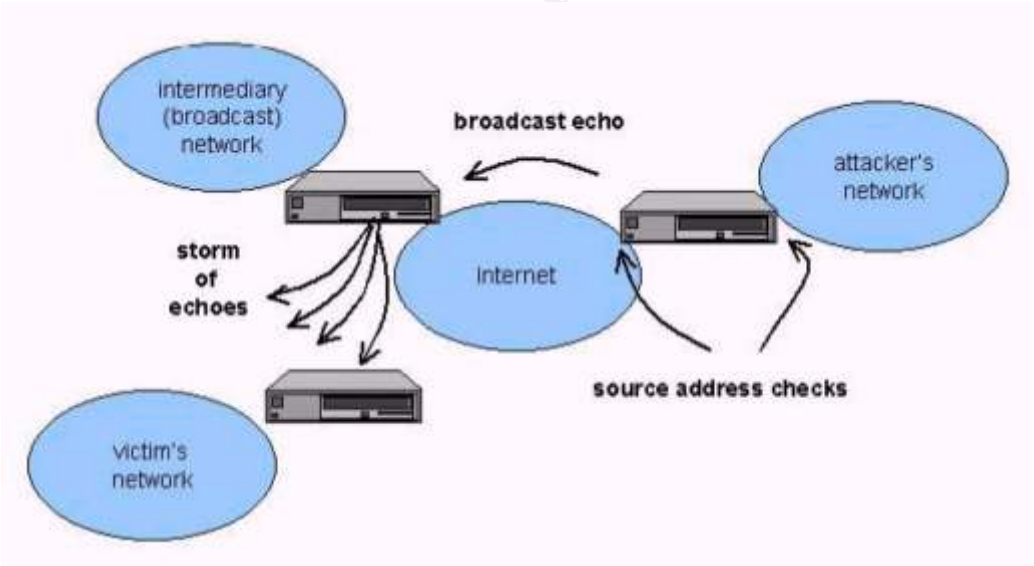
Step 5. Denial of service attack is also possible by not forwarding any packets onto the correct subnet.

ICMP Floods

By flooding the target host with great amounts of ICMP messages will leave the attacked host and its associated network with degraded performance or even total denial of service in some instance.

Smurf (http://cs.baylor.edu/~donahoo/NIUNet/hacking/smurf/smurf.c) attacks are clever: They use whole networks of computers to direct an overwhelming amount of traffic to a victim's machine and its network.

A smurf attack is illustrated below:




Step 1. Attacker finds some intermediary network that will respond to the network's broadcast address.

Step 2. Attacker spoofs the IP address of the victim host and sends a great number of ICMP echo request packets to the broadcast address of the above intermediary networks.

Step 3. Now all the hosts on that network will respond to that ICMP echo request with a corresponding ICMP reply request back to the spoofed IP address (the victim).

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

Step 4. This will send a whole bunch of ICMP echo replies to the victim and its network thus causing network degradation or a total denial of service.

Phase 3 - Keeping Access & Covering The Tracks

After an attacker has successfully compromised a system, one of the ways to hide information as it is being transmitted across a network is to use a technique called tunneling. Tunneling involves hiding one protocol inside another protocol. Loki2 is one such implementation discussed in (http://www.phrack.org/show.php?p=51) which uses ICMP and UDP protocol tunneling to obtain a reverse shell from an attacked system.

The steps to using Loki2 is illustrated below:

Step 1. Attacker gets root on a victim system.

Step 2. Attacker gets Loki2 and compiles it on the machine.

Step 3. Attacker now launches Loki2 client on the attacking machine and gets a reverse shell on the victim host.

Step 4. Now attacker has shell access to victim's machine while tunneling traffic through normal ICMP data packets.

In such an attack, the traffic that is being exchanged between the Loki client & Loki server is almost covert as there are no listening ports opened on the victim machine and even the traffic could be encrypted with an encryption algorithm like Blowfish or DH for additional covertness.

Loki2 when implemented as a kernel module would be even stealthier as it would not even have a process that will sit and wait for the ICMP traffic that can potentially be detected by an alert administrator.

Taking stock of the recent Distributed Denial Of Service (DDOS) attacks, we have seen that ICMP have been used in almost all of those tools for covert communications between the DDOS client and the attacker's handler program. Few examples are TFN2K and Stacheldraht.

**Conclusion**

We have seen throughout this paper that ICMP can and has been used in many phases of an attacker's advance in a system compromise. In many instances, tools are easily available on the Internet for download.

We've also seen that ICMP is not just being used in the reconnaissance & scanning phase which is most understood but it has also been used for exploiting systems as well as in certain instances as a covert channel for attacker's communication.

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

**References**


1. Madalina Baltatu, Antonio Lioy, Fabio Maino, Daniele Mazzocchi. "Security Issues in Control, Management and Routing Protocols". 22-25 May 2000. URL : http://www.terena.nl/tnc2000/proceedings/3A/3a2.pdf (10 Dec 2001).

2. "Internet RFC/STD/FYI/BCP Archives - RFC791" URL : http://www.faqs.org/rfcs/rfc791.html (10 Dec 2001).

3. Alex Peeters. "ICMP Header Format". 4 October, 1999. URL : http://citap.freeservers.com/publications/tcp-ip/tcpip012.htm (10 Dec 2001).

4. "Nmap" URL : http://www.insecure.org/nmap (10 Dec 2001).

5. Foundstone, Inc. "Superscan v3.0". URL : http://www.foundstone.com/rdlabs/proddesc/superscan.html (10 Dec 2001).

6. Mark Spencer. "Cheops". 1999. URL : http://www.marko.net/cheops (10 Dec 2001).

7. "Firewalk" URL : http://packetstorm.decepticons.org/UNIX/audit/firewalk/ (10 Dec 2001).

8. David Goldsmith & Michael Schiffman. "Firewalking". October 1998. URL : http://www.packetfactory.net/Projects/Firewalk/firewalk-final.html (10 Dec 2001).

9. Ofir Arkin. "X Remote ICMP based OS Fingerprinting techniques" August 2001. URL : http://www.sys-security.com/archive/papers/X_v1.0.pdf (10 Dec 2001).

10. Fyodor Yarochkin & Ofir Arkin. "X Probe tool". URL : http://www.sys-security.com/html/projects/X.html (10 Dec 2001).

11. Antirez. "Hping2 tool". URL : http://www.securityfocus.com/tools/641 (10 Dec 2001).

12. "SSPing tool" URL http://packetstormsecurity.org/Exploit_Code_Archive/ssping.zip (10 Dec 2001).

13. "Jolt" URL : http://www.wwdsi.com/demo/saint_tutorials/jolt.html (10 Dec 2001).

14. G P R . "TearDrop tool". 13 Nov 1997. URL : http://packetstormsecurity.org/Exploit_Code_Archive/teardrop.c (10 Dec 2001).

15. Lindsay van Eden. "The Truth About ICMP". 17 May 2001.


© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

URL : http://www.sans.org/infosecFAQ/threats/ICMP.htm (10 Dec 2001).

16. Pete Schuyler . "Getting More Out of ICMP". 16 May, 2001
URL : http://www.sans.org/infosecFAQ/audit/more_ICMP.htm (10 Dec 2001).

17. S. Deering. "Internet RFC/STD/FYI/BCP Archives - RFC1256". September 1991.
URL : http://www.faqs.org/rfcs/rfc1256.html (10 Dec 2001).

18. Craig A. Huegen. "Smurf Information". 7 February, 2000
URL : http://www.pentics.net/denial-of-service/white-papers/smurf.cgi (10 Dec 2001).

19. TFreak. "Smurf.c".
URL : http://cs.baylor.edu/~donahoo/NIUNet/hacking/smurf/smurf.c (10 Dec 2001).

20. Mark Gibbs. "Attacked by Smurf". February 22, 1999.
URL : http://www.nwfusion.com/archive/1999b/0222gearhead.html (10 Dec 2001).

21. "Denial Of Service Attack Swords". May 2000.
URL : http://fravia1.virtualave.net/fraviamirror/dod1.htm (10 Dec 2001).

22. daemon9 & alhambra "Project Loki: ICMP Tunnelling." Phrack Magazine, Volume Seven, Issue 49, File 6 of 16, August 1996. URL:
http://www.phrack.org/show.php?p=49&a=6 (10 Dec 2001).

23. daemon9. "LOKI2 (the implementation)." Phrack Magazine, Volume Seven, Issue 51, Article 6 of 17, September 1, 1997. URL:
http://www.phrack.org/show.php?p=51&a=6 (10 Dec 2001).

24. Craig H. Rowland. "Covert Channels in the TCP/IP Protocol Suite" 14 November 1996. URL:
http://www.psionic.com/papers/covert/covert.tcp.txt (10 Dec 2001).

25. David Dittrich. "The Tribe Flood Network distributed denial of service attack tool" 21 October 1999. URL :
http://staff.washington.edu/dittrich/misc/tfn.analysis (10 Dec 2001).

26. David Dittrich. "The stacheldraht distributed denial of service attack tool" 31 December 1999. URL :
http://staff.washington.edu/dittrich/misc/stacheldraht.analysis (10 Dec 2001).

27. Eric Cole. Hackers Beware. New Riders, 2001. pp 182-188, 191-195, 211-217, 229-235, 601-604.

© SANS Institute 2001, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: June 18th, 2013

# Upcoming SANS Training

**Click Here for a full list of all Upcoming SANS Events by Location**

| | | | |
|---|---|---|---|
| **Digital Forensics & Incident Response Summit 2013** | **Austin, TXUS** | **Jul 09, 2013 - Jul 16, 2013** | **Live Event** |
| **SANS London Summer 2013** | **London, GB** | **Jul 09, 2013 - Jul 16, 2013** | **Live Event** |
| **SANS Rocky Mountain 2013** | **Denver, COUS** | **Jul 14, 2013 - Jul 20, 2013** | **Live Event** |
| **SEC528 SANS Training Program for the CompTIA® New Advanced Security Practitioner (CASP) Certification** | **Washington, DCUS** | **Jul 22, 2013 - Jul 26, 2013** | **Live Event** |
| **SANS Mumbai 2013** | **Mumbai, IN** | **Jul 22, 2013 - Jul 27, 2013** | **Live Event** |
| **SANS San Francisco 2013** | **San Francisco, CAUS** | **Jul 29, 2013 - Aug 03, 2013** | **Live Event** |
| **SANS Boston 2013** | **Boston, MAUS** | **Aug 05, 2013 - Aug 10, 2013** | **Live Event** |
| **SANS SEC 560: Network Penetration Testing @ Bangalore 2013** | **Bangalore, IN** | **Aug 05, 2013 - Aug 10, 2013** | **Live Event** |
| **Critical Security Controls Summit** | **Washington, DCUS** | **Aug 12, 2013 - Aug 18, 2013** | **Live Event** |
| **Industrial Control Systems Security Training - DC** | **Washington, DCUS** | **Aug 12, 2013 - Aug 16, 2013** | **Live Event** |
| **SANS Virginia Beach 2013** | **Virginia Beach, VAUS** | **Aug 19, 2013 - Aug 30, 2013** | **Live Event** |
| **SANS Thailand 2013** | **Bangkok, TH** | **Aug 19, 2013 - Aug 31, 2013** | **Live Event** |
| **SANS Melbourne 2013** | **Melbourne, AU** | **Sep 02, 2013 - Sep 07, 2013** | **Live Event** |
| **SANS Capital City 2013** | **Washington, DCUS** | **Sep 03, 2013 - Sep 08, 2013** | **Live Event** |
| **Network Security 2013** | **Las Vegas, NVUS** | **Sep 14, 2013 - Sep 23, 2013** | **Live Event** |
| **SANS Canberra 2013** | **OnlineAU** | **Jul 01, 2013 - Jul 13, 2013** | **Live Event** |
| **SANS OnDemand** | **Books & MP3s OnlyUS** | **Anytime** | **Self Paced** |