Global Information Assurance Certification Paper

FORWARD BY STEPHEN NORTHCUTT:

The SANS/GIAC IDIC Grading panel asked me to provide guidance in the scoring of this submission. In some sense it is outside the guidelines of the assignment, to report and analyze ten detects. In the end, all GIAC practical scores come down to two questions, did the student demonstrate knowledge of the subject matter and does the submission advance the state of practice in defensive information operations. This passes those criteria heartily, I am only authorized to issue the first, or draft grade, but while the power is mine: 100 *

PURPOSE:

This analysis report is part of the SANS GIAC IDIC LevelTwo certification program, and satisfies the practical portion of that certification curriculum requirement. The assignment called for the analysis of 10 observed attack signatures, but this report goes beyond this mandate, examining a nearly uninterrupted 60-day collection of Cox@Home 24.6.137.x sub-network traffic. The activity and anomalies recorded by NetworkIce "BlackIce Defender" (BID), installed in "paranoid" mode, provided the basis for attack signatures seen. The Detection descriptions provided below expand upon the NetworkIce advisory text that helps understand the BID identified "Issue Names" found in the evidence log files. The content of the report describes the detections made by an arbitrary classification scheme developed for the purpose of handling many detections of similar characteristics, and for using this aggregation to better understand the severity and risk associated with each. A generalized description of each class of attack is presented followed by the significance of each signature seen. Those felt to be most useful were traced using the WinNT *tracert* package where an attacker's identity could be better verified, or where other data might be discovered for correlation. BID evidence log files and my Summary of Detections are provided separately.

SCOPE:

The scope of this report encompasses intrusion detections accomplished on a nearly continuous basis from Tuesday, at 12:15UTC February 15, 2000 though Monday, 16:29 April 10, 2000, except for two gaps. One 2:27 minute gap from 19:50 to 22:17 on 4/2/2000 resulted from a change in log file operations as the software for BID was upgraded to 1.9.25. The other gap of 8:04 on 4/3/2000 from 06:36 until 14:32 was due to log file capture and output to support a Phoenix ISSA briefing on Intrusion Detection. With the exception of the total of these 10:34 gaps, this effort amounts to an uninterrupted 60 day baseline of information. This represents a valid sampling of the "live" threats witnessed and collected from my Cox@Home 10MB network. The supporting raw traffic file history from 4/5/2000 through 4/10/2000 is archived pending resolution of the reported stack overflow/access violation problem with the CiALL protocol decoder tool, or configuration and operation of the www.netgroup-serv.polito.it/Analyzer protocol decoder from Turin Polytechnic. This is underway at this time to develop the forensic log files necessary for incident reporting to Abuse@Cox.com.

NOTE:

The following explanatory comments in italics, copyrighted by Network Ice in their online KnowledgeBase notes, are reproduced with credit through their courtesy for better understanding and comprehension of this SANS IDIC Certification Practical Report. *The columns* in the attached Excel spreadsheet attack-lists *are* identified, from left to right:

Severity - This is a number from 1-99 that indicates the severity of an attack, where 1 is not very severe, and 99 is the most severe attack. Unfortunately, these levels do not have any precise meaning. Even an attack at level 1 may result in a compromise of the machine, whereas an attack at level 99 could be harmless. The assigned level is just a best-guess.

Timestamp - This indicates the time and date of the last time the attack occurred. Attacks are "coalesced", meaning that if the same attack occurs multiple times, earlier attacks are sometimes removed from the list and simply merged with the latest one. A count of the number of times an attack has occurred is kept in

another column. This timestamp is kept in GMT (aka UTC), and is probably several hours off from the time you see in the user interface. The ISP will want the time in this format so they don't have to worry about what timezone you are in.

"IssueName" - The name of the attack. Each of the unique "issueId" numbers has a name associated with it.

Intruder's IP address - The IP address of the attacker. Remember that IP addresses can sometimes be "spoofed" (forged), or that an intrusion may be a "false-positive", so there isn't a 100% chance that this is actually a hostile person.

Intruder's name - The name of the intruder. We scan both Internet databases like DNS as well as the attacker itself in order to find the "best-name" of the machine, then display it here.

Victim's IP address - This is the IP address of who the intruder was attacking. For example, if a user is running the product and gets attacked on a dial-up, then this will be the IP address assigned to that machine during that dialup session.

"Parameters" - This contains some detailed information about the attack. For example, in a "TCP port probe" scan, this will contain a list of "ports" the attacker was scanning. The meaning of this information is documented in the "advICE" database.

Count - The number of times this attack was seen.

DETECTION ANALYSIS:

The descriptive segments and analyses of each detection class follow the Information Warfare approach recommended by the SANS Institute. Each segment attempts to characterize each class of detection by their attributes, according to the following criteria:

Existence – identify hostile individuals and groups
History – identify their history
Techniques – evidence of intent?
Intent – Evidence of active targeting?
Targeting - Evaluate information

Excel spreadsheet formatted outputs of the evidence evdxxxxxxxx-xx.log files are attached separately, with an ASCII listing of the raw traffic logxxx.enc files inventoried for decode and submission to Cox. An accompanying Summary of Detects by Category is provided in the one page Excel spreadsheet, identifying not only the specific scans/probe and map/fingerprint attacks witnessed, but also showing their distribution over the 8 weeks of monitoring. Scans believed to originate from Cox for network management purposes are identified as non-hostile, with the notation that they were apparently sourced by either Cox operations or Cox customer service ("Whats Up scan"). Specific actions seen have been classified below into the categories shown and track to each in order of appearance on the Summary of Detects by Category:

Summary of Multicast/Discovery scans: These scans are recognizable by the netmask 255.255.255.255 and reflect potentially harmful network reconnaissance activity.

Multicast scans: Two of the Internet Communications Message Protocol (ICMP) delivered scans were seen, one of which was identified by its use of a multicast address (255.255.255.255) for both source and destination IP addresses. The other ICMP scan was sourced at 224.0.0.7, and the name lookup done by BID returned "ST-ROUTERS.MCAST.NET" as the source DNS. As these were both detected at exactly the same time, they are felt to originate from the same source. "Multicast address 224.0.0.4 is a well-known address used by DCMRP (Distance Vector Multicast Routing Protocol), the protocol currently used for multicast routing. (DVMRP is defined in RFC 1075 [Waitzman, Partridge, and Deering 1988]."[1] However, the protocol identified by BID is ICMP, not IGMP. This is very curious at it appears to qualify as some sort of net-directed broadcast because the mask is not sub-netted. Were this in fact a subnet-directed

broadcast, then its use of ICMP would be better appreciated. But, it is constructed as a network-directed broadcast, except that it does not appear as IGMP traffic, but rather as ICMP. The intent under the circumstances as given and witnessed is unknown and suspect, and quite possibly something less than benign as there is clearly no capability to perform Network Management services across the Cox network (excepting, of course, Cox@Home). Some insights and thoughts on this would be appreciated from my Cox@Home provider, and these will be requested, for the record!

[1] TCP/IP Illustrated, Volume I, The Protocols, W. Richard Stevens, Addison-Wesley Longman, Inc., 1004, pg. 185.

SNMP discovery scans: These were separately classified and reported herein on the basis of their origin inside/outside the Cox domain. However, source IPs 209.220.106.220-225 seem to be regularly involved in the execution of these scans. Also, 192.168.x.x seems to account for a large portion of the other SNMP scans seen from outside the Cox domain (24.6.x.x). As 192.168.0.1 is but one of these source IP addresses, it maybe speculated that the private 198.168.x.x address used to obscure, obstruct, and complicate traceability, and thereby diminish its forensic value.

As the use of SNMP Get requests over UDP for network management is presumed to be limited to internal network administration agents/people, it is felt that seeing these from outside the enterprise confirms the intent as non-benign in character. At the best it could constitute leakage into the network neighborhood, and intelligence gathering prior to an attack at the worst. In addition, what could be interpreted as SNMP community string guessing is seen on Feb 24, 2000, where four external SNMP scans are recorded as having used both “internal” and “public” community strings are part of an attempt to gain access and enumerate hosts for identification information.

In addition, the internal SNMP discovery scans have also been seen to use these “internal|public” SNMP community string passwords in an attempt to do the same thing. For these reasons, these internal 24.x.x.x SNMP sources are also suspect as representing internal and potentially hostile probing as well.

Summary of Network Management scans, probes, pings: These scans represent the activity of folks at Cox who are responsible for network management operations and customer service on behalf of paid subscribers. This category includes all events that are believed to fall within the scope of authorized maintenance and service personnel who are Cox employees or contractors. *NOTE: One late discovered hostile SMTP port probe was determined and written up as such, but was not relocated for separate accounting.

WhatsUp scan: The Network Management toolset that is apparently is use on *Cox@Home* is the WhatsUp product from IPSwitch (*www.ipswitch.com*). This tool appears to be in use by Cox Customer Service to evaluate the status of subscribers throughout one or more tiers, especially as it is far cheaper than HP OpenView. BID found that the name of the client initiating one of the 2 instances recorded was that of “worktsr-13.custops.home.net.” The only other case also came from “cj753197-a.alex1.va.home.net” within the Cox domain and may reflect authorized use, especially as it was seen only.

SMTP Port probe: This normally means that someone is scanning the system to see if it supports the Simple Mail Transport Protocol (SMTP) mail transfer service. In the first case seen on 2/24/00, the probe originates from *Cox@Home.com*. However, in the second* case, on 4/3/00, it does not. The source IP address used is 0.0.0.0 and no BID alternative lookup for the host name was successful. This is indicative of a non-benign probe. That this SMTP probe is see at exactly the same with the same cloaked identity is further confirmation of the hostile and stealth attempt to gain information about the DNS services on the 24.1.240.x subnet, as well as the mail services on the 24.14.30.x subnet. Normally, when the email client is configured, the ISP email server is identified as the address where email is directed on the outbound side, and from where it is received by the client on the inbound side. The first detect reveals an authorized source and intent. The second* actually represents one of the biggest problems on the Internet, where spammers scan the Internet looking for mis-configured SMTP servers. Their intent is to locate an address that they can “cloak” their true email identity. By forwarding outbound e-mail in this manner, spammers can "anonymize" their connections and hide their tracks, avoiding the inevitable

customer complaints to their ISPs that result in their being "blackholed." This would take them off many approved site address lists and greatly diminish their access to Internet email web sites. More information on "blackhole" methods and techniques to current threats can be found at: *http://maps.vix.com/rbl/candidacy.html*. This is the more extreme of the conditions they face, as merely installing a filter on your email client serves to label it as an offending address, causing the email from it to be dropped. In addition, and far more importantly, spammers can use a relatively low-speed dialup modem link to this to hijack you email server and send bulk email traffic over high-bandwidth connections.

NNTP Port probe: Network News Transport Protocol (NNTP) is the used to transfer USENET messages around the Internet. Hackers regularly scan the Internet looking for machines that might support this service. Primarily, they are looking for machines they can be used to anonymously read messages from or post (write) messages for reading by others. Probing known NNTP ports enables someone to find systems where these services are enabled and determine if these web sites are "open" NNTP servers, meaning that they can be used anonymously. This is the ideal situation for attackers as it keeps their identity hidden. The NNTP probes seen during this 2 month period all appear to be authorized as the names returned by BID would seem to indicate they originate from within the *Cox@Home* domain.

PCAnywhere ping: This detection indicates that someone has pinged systems on the *Cox@Home* network to see if PCAnywhere is running on any connected hosts. This may be an attack, but is likely to be accidental. PCAnywhere is a product from Symantec allowing remote access control of a computer. It is very popular on the Internet for this legitimate purpose, permitting administrators to perform distributed network management of computing resources. However, "Black Hats" also know its value and frequently scan the Internet looking for machines supporting this product. Many users use "null" password strings or passwords or easily guessed strings ("password") for quick and quiet entry. The intent is use this often unchecked "backdoor" capability, often relying upon unauthenticated modem dialup session connections, to subvert your OS and gain control, discovery account identification and password data stored on the local hosts. Theft of this kind of information often enables further break-ins across the Internet, falling as in the "domino effect." Accidental scans from PCAnywhere clients are commonly seen from your network neighbors due to the default configuration of Pcanywhere, as it installs an icon called "NETWORK" that scans the local area for agents. Though rude and ill-mannered as this may be, there is no clearly hostile intent behind this scan just because it is detected. So, if this action is seen from within your network domain, it is probably benign, and clearly non-benign is it is seen to originate from outside *Cox@Home*, in this case.

All the PCAnywhere pings seen were attempting to determine the status port UDP 22, and most except one also tried to determine the status of UDP 5632 as well, as this port is used for the PNAnywhere application for UDP data flow. These attempts were launched to determine is the host was running the application, and to see if the service application, if installed and configured, was busy of free. All attempts to determine this status were blocked by BID, despite the fact that PCAnywhere is not installed on the host being scanned.

Summary of Port Scans: These scans represent hostile activity initiated primarily, but not exclusively, from outside the *Cox@Home* domain. Scans are done to determine which services are enabled and hopefully "open" for misdeed usage. Some of the more insidious scans shown in red represent deliberate and rather exhaustive attempts to locate known Trojan program communication ports for covert channel exploitation. Some of these channels are for remote access and stealth command delivery inside firewalls, and others are for receiving covert channel outflow, as is the case of the EvilFTP scan witnessed.

TCP Port scan: "Black Hat" folks like to scan to find out what services you system uses that are configured for use, and whether they may attacked with techniques and tools available from download from hacker websites. Once the services are identified it then only becomes necessary to find out the versions of the software running on your system that support these services. From other information sources the attacker gains knowledge of current vulnerabilities within the software supporting these versions. So, when the attacker knows what services you use, he probably already knows the ones he is prepared to target for exploitation. If true, the attack planning stage may be over, and the next sound you hear is ***CRASH***.

Specific ports detected doing this type of scan include are listed in the three groups below for further examination:

a. 7, 20-2, 25, 35, 37, 42, 53, 69-70, 79-80, 106-7, 109-111, 119, 143, 1745, 2301, 5190-93, 5631-32, 5800, 5900, 6000, 8000, 8010, 8080, 9100, 25867;

b. 27267, 33895, 34065, 34816, 36101, 37126, 37962, 39017, 39380, 39823, 40147, 40485, 41926, 41963, 42619, 44010, 46404, 47979, 49708, 49774, 50535, 51254, 51315, 51326, 51838, 52378, 54061, **54320**, 55996, 56144, 56401, 56575, 57155, 58670, 61379, 61418, 62300, 62441, 64181, 64291, 64930; and

c. 31, 555~2023, 2565~2801, 4950~5742, 6400~7306, 9872, 11000~11223, 12076, 16969, 20000~20034, 21554, 22222, 23456, 30100~30303, 30999~31337, 34324, 40412, 50766, 53001, 61466, 65000.

The ports identified in subparagraph a., above, in light blue, are those that were scanned. These scans are considered to be relatively benign, at least compared to the ones that follow. This determination is subjective and is based principally on the fact that they appear to originate from within the *Cox@Home* domain and the ports involved are confined to those well-known ports defined by the IANA. That there is no good reason these should even be seen in the first place adds a suspicious flavor, and compels the paranoid nature of security folks to remain circumspect about true intent until more information is held.

The ports scans identified in subparagraph b., above, in light green, are those that appear to originate from outside the *Cox@Home* domain. The IANA ports involved are well into the range of ephemeral ports typically shared with temporary client service requests. That these scans are specifically looking for ports in the range(s) reported by BID (nnnnn~nnnnn+1) appears to indicate that may exist an intended use for the ports which is not shared with the IANA. If so, this could mean that the ports may have potential usefulness for hackers. If these ports are not assigned and are not in use, then they could be recruited as non-default trojan ports for modified or new exploits. It is also noteworthy that these particular scans came from a range of source IPs that are also associated with an ensuing series of Trace route and Echo Storm activities described later. If not indeed reconnaissance activity, these scans may be linked to the brace of detects seen from roughly 00:30 to 02:30 UTC on 3/16/00. This is too coincidental for mere chance, and the recurrence of 63.226.21.232 and 63.226.21.239 is too great to be ignored. Some form of coordinated attack activity is felt to be the cause of this, though the specific attack profile is indeterminate at this time. One last observation is that the port bolded and underlines above, port 54320, is identified by *www.doshelp.com/trojanports.htm* as a UDP trojan port for Back Orifice, and is very close to the default TCP port 54321, shared by School Bus and Back Orifice.

The ports identified in subparagraph c., above, in red, are also from outside the *Cox@Home* domain, but these are obvious scans for trojan ports of known identity, using the specific names "EvilFtp," "GateCrasher," and "NetSphere." There can be no doubt that these scans have a hostile intent, as they are ALL listed on the *www.doshelp.com/trojanports.htm* reference page, from Agent 31 / Hacker's Paradise (31) to Stacheldraft (65000).

TCP "FIN" scan: "Black Hat" folks like to find out if they can connect to the system without really connecting, and this method is a preferred one because it is rather "quiet" and often escapes ready notice. In this case, the attacker is using a method called a "FIN" scan. It attempts to close a non-existent connection on the server. Either way, it condition is erroneous, but the different Operating Systems (OSs) tend to respond differently, some with an intent to be helpful. This is often the result of the each specific OS implementation of the TCP stack. Often the desire to be helpful errs on the side of providing more than sufficient information than is actually necessary. When the person/agent behind the request is more than merely inept or uninformed, it can result in the flow of enough information to permit the OS to be positively identified for subsequent attack targeting. FIN scans also tend to avoid normal logging triggers set up on the system, unless an intrusion detection system is in place, or unless a firewall with dynamic filtering capability is configured and used to provide a similar "state-based" traffic analysis capability.

One externally launched scan was seen during this entire period from 203.251.87.5 as the first of 4 uniquely different scans done within the space of five seconds (see also the TCP "ack" ping described below). One additional scan of a TCP port probe nature concluded this scan series and it occurred ten

minutes later, focusing on port 22. The FIN scan itself ran against ports 22 (PC Anywhere), 53 (DNS), 2049 (unknown), and 2222 (unknown). If the attacker finds anyone listening to port 22, it means they are running older versions of PC Anywhere and a PC Anywhere-ping should be expected to follow shortly afterwards. PCAnywhere listens on ports 22 (TCP and UDP), 5631 (TCP) "pcanywheredata", and 5632 (TCP and UDP) "pcanywherestat), and 65301 (TCP). It uses an "IP discovery protocol" to find other PCAnywhere servers on the local segment, where the assumption is that the local segment is all IP addresses between "xxx.xxx.xxx.1" to "xxx.xxx.xxx.254" (i.e. the local class C allocation). Thus, cable-modem and DSL users will often see connections to this port from other people that have PCAnywhere installed. However, in this particular case, in light of the other scans also seen, this is no accident from someone running PCAnywhere and seeking to connect for authorized service access.

Summary of Pings: These BID detected network scans are slightly less intrusive in nature than probes, and have the property of disclosing the existence or non-existence of certain services on network host machines. It is not necessarily a quiet method for scanning, nor is it associated with the actual break-in phase of any attack. It does reflect that someone is looking at you, and trying to size your machine up as a possible meal.

Ping Sweep: "Black Hats" rely upon this technique to scan an entire network looking for systems that are alive and ready for attack. The ICMP echo request "ping" is swept across a target network to locate any hosts that respond with an echo reply, indicating they are alive. As one of the most classic hacker activities on the Internet, the hacker scans the network looking for machines that are alive, a necessary condition if they are to be attacked! Network management systems sometimes send these requests, but under the circumstance in which these 3 pings were seen, it is safe to say they were not coming from any identifiable *Cox@Home* name source, and so these must be considered to be hostile in intent.

TCP "ack" ping: Two instances of this TCP ack ping were seen. The first originated from within the *Cox@Home* domain and could well be benign, especially as the source IP address 24.8.69.127 used the subnet mask 24.8.69.255. This is quite possibly a condition caused by network congestion rather than an intrusion. However, the second instance detected originated from outside the *Cox@Home* domain and appears to be attempt to run a stealth scan against the network. This attempt was the second in a series of 4 quickly executed scans against my host (see TCP FIN scan above). The fact that the BID firewall features on this host (my computer) are set to automatically block incoming connection attempts, like this, after only the first few frames of a connection, is evidence that a crafted packet attack was effectively blunted without responding to the ping attempt. Had this hostile ping been answered, it is most certain that additional efforts would have been made to use the knowledge gained. However, this scanning would not have directly placed my system at risk of compromise, but it would have served to entice the attacker that there was something behind my firewall that might be worthy of further investigation and an actual break-in attempt. That no further activity has been seen from 203.251.87.5 since 3/14/00, the date of this TCP ack scan was detected, it can be safely assumed that this "Black Hat" individual is off chasing some less well-defended site.

Back Orifice ping: Back Orifice (BO) pings are the most frequent attack seen on the Internet. Having detected and logged this TCP based ping without responding to it, means that my host machine has been successfully protected by BID, as no response from my machine was elicited. This means that I have been scanned, but not targeted. Hackers scan thousands of machines on the Internet hoping to locate those that have been infected by a BO trojan implant. Because this protected host is also protected by virus-scanning software to detect and remove Bo and other Trojan files, the BID firewall/intrusion system can concentrate on traffic monitoring to nail any further attempts to use the trojan code had it been successfully implanted. This double layer of protection constitutes protection in depth. It is also noteworthy that two variants of BO were detected and interdicted by BID. The first variant seen, and the only instance of it as well, was the BOWhack attack.

The BOWhack originated from source IP 203.32.78.128 and was named "rich128.ozdocs.net.au". The BOWhack distinction made was done based on the BID parameter decode data **(bolded)** seen:

type = the Back Orifice command (**PING**, SYSINFO, PROCESSKILL, etc.)

passwd_hash = the hash of the password used to encrypt the Back Orifice traffic. While it is impossible to determine the original password used to generate this hash, knowing the hash can sometimes help differentiate between "script-kiddies" who use the default password (with hash of **0x7A69**), and serious crackers who use a non-default password.

length = the length of the Back Orifice packet (**198**).

xid = the XID is often **0x0** for sweeping programs, or some other value for the BO Client. The former indicates the attacker has no specific interest in the target machine; the later is a good indication the intruder is interested in finding Back Orifice on that one machine.

vport = victim's port number (the port the BO server is running at). A port of 31337 is the default port, and indicates activity by "script-kiddies", while other ports **(31666)** indicate a serious cracker is targeting the system.

iport = intruder's port number. Different well-known BO clients use different port numbers **(1764).**

The remaining eleven Back Orifice detections seen were all instances of the same standard Back Orifice (BO) variant, using default vport number **31337**, a uniform xid value of **0x0**, and a uniform length of **188** bytes. With one exception, the iport values used varied with each attack signature, as one might expect since each attack originated from a different source IP address. The one exception noted involved the use of a very large xid, whose value was consistently 0xFFFFFFFF (94967295), and whose iport value was also very large, using 94950977 (0xFFFC041) the first time, and 94940289 90xFFFF9681) the second time. It is curious that the same source IP did not send both of these BO pings with the large iport number.

Summary of Port Probes: The following port probes comprise a more dangerous category of hostile port activity, especially as viewed towards known tropjan covert communication channels. Probes have the potential to result in a computer break-in, either directly during or indirectly after the probe is completed.

PCAnywhere Port probe: In addition to the PCAnywhere ping described earlier, there is another PCAnywhere application port that was actually probed for availability and use. This additional port is TCP port 5631, the one actually used for remote control over the host executing the PCAnywhere application. That this request should arrive across the *Cox@Home* network to my computer, when I am the sole Administrator, without my authorization, is clearly a hostile act. BID is configured to block any such requests automatically, and this did result in the attack being blunted, so that the probe attempt was silently dropped. It is interesting to note that this attempt originated from source IP 24.95.166.101 with the name "gdh2-265.twcny.rr.com" and that it was never seen again at any time during the period of monitoring.

FTP Port probe: Receiving this notice means that somebody tried to access my machine using the FTP protocol and failed. This type of Internet intrusion detection is common because "Black Hats" frequently execute broad network wide scans to find FTP servers. The results of such a scan can reveal thousands if not millions of machines that offer this very common service. Seeing this at a host protected by BID is no cause for alarm, especially as the system that reported this FTP scan is not configured to operate as an FTP server so configured. FTP is one of the oldest protocols on the Internet, dating to the 1980s, making it the granddaddy backbone for data transport and information sharing, long before the advent of HTTP which has largely supplanted FTP today. That the HTTP service supports passive FTP affords it added protection capabilities over active FTP. FTP's persistence is due to the improved efficiency it affords to data exchange, specifically because of the method it uses to transport files. Probing for FTP server availability is done for several reasons:

1. To break into your machine by attacking the FTP server itself. (See FTP Exploits for a description of some of these attacks).
2. To find a covert storage channel/warehouse ("drop-off point") site for exchanging illegal files (warez, porn, MP3s, pirated software), particularly because the size of these files is so large and related access traffic to these sites so great that it would tend to betray there true identity to law enforcement.

3. The efficiency of file transfer, particularly over wideband networks, is advantageous as well for the purpose of cloaking the presence and true nature of the data transfers.

FTP is an excellent way to open up a server to which other people can upload files. In order to guard against hackers using your server as a drop-off point, you must make sure that the directories that are intended to be read, are set with read-only attributes, to prevent them from being written into by external FTP service consumer clients. The exact method needed varies with the way FTP services are configured for use by the OSs on which they run, but the control features (Unix ownership and permissions, or Windows ACLs) all support this capability.

The specific instances where this FTP port probe was seen on my host NT workstation are merely anomalies as a result of the installation of BID and the fact that my host uses promiscuous mode to observe other traffic on the *Cox@Home* network. BID is operated on this site under a non-commercial license and without intent to support FTP service. When BID was installed it automatically blocked FTP service. BID has a configuration option to enable FTP service to work through a firewall, but this option is not used for security assurance.

One of the external *Cox@Home* probes appeared to originate from Dokuz Eylul University in Izmir, Turkey, which an NT tracert revealed to be sourced at 193.140.152.32 with a DNS name of "teos.iibf.deu.edu.tf," tending to confirm that this is a Turkish university site. The second external FTP probe was from "21-25-212-172.san.rr.com" which an NT *tracert* revealed was actually sourced at 24.25.192.58 with a DNS name of "mcr3.san.rr.com," which is a RoadRunner high bandwidth ISP site, probably in San Antonio, TX. A third FTP probe was investigated, and was NT *tracert*'ed to the DNS name "arcano.cic.userena.ci," but the "Unable to resolve target name….arcano.cic.userena.ci." message was returned. Because there was no intent to become too curious and pushy, no reverse DNS lookups were initiated using any IP addresses discovered during this or any other investigation phase of the analysis conducted!

UDP Port probe: UDP port probes may be deliberate "Black Hat" attempts to find an exploit a UDP connection. In this case they are seen as positive detections of intrusion. But they may also be false-positives! When a user first dials up to the Internet, busy ISPs will often use Dynamic Host Configuration Protocol (DHCP) to recycle IP addresses. It does this through "pooling" or sharing of the same IP addresses. Allocation of an IP address under this scheme is dynamic, and as each connection is requested, the next available pooled address it re-assigned. As each switched connection is closed (call completed), the IP address is re-allocated and returned to the free IP address pool. As ISPs quickly turn over IP addresses assignments, the real-time timing delay experienced from the closure completion, and the completion of pooled address housekeeping updates, to the next open request initiation can cause this mis-routing "side affect" to occur. The result is that the UDP port probes typically use UDP/TCP fort 427 for server location. BID fires this detection alert whenever the host computer receives UDP data it knows it did not request, and all the ones seen to occur between 2/29/00 and 3/5/00 were of this kind. The first 10 detects also seen from 4/3/00 to 4/7/00 were also false-positives as they were attempting to use the well-known UDP port 513 that maintains data bases showing who's logged in to machines on a local network, as well as the load average of the machine. Four more were seen later in the day on 4/7/00 against port 513, as were another on 4/8/00 and two on 4/10/00.

Other UDP port probes against the following ports were also seen, and identified by IANA assignment, in parentheses, and for this reason were considered benign: 3283 (Net Assistant), 4000 (Terabase).

However, the UDP port probes against the following ports were also seen, but were not identifiable by IANA assignment, or RealAudio (6970-7080) audio/video servers, or any other known benign or hostile application/service: 62079 , 61333, 1031-32, 1042, 61881, 62154, 39213(2). Their intent can only be speculatively thought to be a survey of unused UDP port for future hostile use. Based upon further information from NetworkIce, to the affect that DSL and cable-modem subscribers should expect at least one UDP port scan per day due to being always-on, I would say that the rash of UDP port scans seen in the evidence log attached for 4/8/00 far exceed that number.

Considering that on 4/8/00 at 06:41:58 there were a total of 26 rapid succession UDP port scans directed against 24.14.31.224 port 1031-32 and 1042, it is too difficult to chalk this up to randomness in any sense. Assuming the source addresses are spoofed, as per the additional commentary (below in italics) from NetworkIce would suggest, it could be argued that this activity reflects a serious, systematic, and overt (non-stealthy) UDP port probe automated sequences, executed much as a fast sweep for current port availability for nefarious use. Or, more remotely, that there are perhaps trojan UDP ports within these ranges that "White Hats" have yet to watch more carefully in order to understand how they are/maybe used. Perhaps something like a potentially armful "watch list" for surveillance of active port usage would be a useful background task within the SANS community (if not already done).

"About 10% of these scans are from forged (spoofed) addresses. This means the indicated IP address in the attack is probably from the real attack, but a small percentage of the time the indicated person is completely innocent. About 20% of these scans are from machines already compromised by a hacker. In other words, if you report this scan back to the originator, they may thank you, because you've discovered a hacked system on their network they didn't know about."

TCP Port probe: BID returns indications of a TCP port probe when somebody tries and fails to gain unauthorized access to your computer using any of the IANA defined TCP ports, or any of the default ports associated with any of the 600 attack signatures defined in the NetworkIce advice database. The TCP port scans previously described above are distinguished from the TCP port probes described here in that the scans are more limited in terms of the scope of activity they do. Scanning is analogous to a thief that cases a neighborhood looking for likely places to burglarize. Probing goes one step further by actually visiting the premises to try the doors and windows (sic) to see if physical ingress into the house is possible. It is viewed as the last step before the break-in phase of a burglary. Probes are more determined and represent a clear intent to do wrong at the time the detection is made and reported to the security administrator of the system. BID is configured on my computer in "paranoid mode" so that is errs on the side of protectiveness, and it considers any unanticipated attempt to access TCP services through IANA defined TCP ports as a violation of the security policy in force. Web servers, ftp servers, audio/video services are all included in this regard.

Specific ports detected doing this type of probe are listed in the four groups below for further examination. These ports were seen being probed from both inside/outside the *Cox@Home* domain:

a. 4286 (IANA assigned this port to VRML Multi User Systems) by source IP 167.216.133.33 (server1.sans.org)

The port probe identified in subparagraph a., above, in light blue, corresponds to that probe is considered to be benign because it originates from one of the trusted SANS servers.

b. 22 (IANA assigned this to SSH Remote Login Protocol) by source IP 203.251.87.5;
c. 109 (IANA assigned this port to POP v2) by source IP 193.164.172.98; and
d. 635 (IANA assigned this port to RLZ Dbase) by source IP 216.119.140.30, 24.1.213.84, and 211.40.249.72;

The port probes identified in subparagraph b-d., above, in pink, are considered to be less benign because there is no good expectation that the need for this probe exists, and although it is targeted against well-known IANA ports, it's occurrence is not felt to be necessary to or linked to any actions from my most computer. In short, there is no need for my host to receive these probes at all. BID makes sure that only those logical responses to requests for service from my host are accepted.

e. 2811 (IANA assigned this to GSI FTP) by source IP 0.0.0.0;
f. 4480 by source IP 0.0.0.0; and
g. 1863 by source IP 0.0.0.0.

The port probes identified in subparagraph e-g., above, in yellow, are those that appear to originate from outside the *Cox@Home* domain. The first probe uses a legitimate IANA port (2811), but this does nothing to alleviate concern that all of these probes are hostile, despite the fact that no specific purpose for them is identified at this time.

h. 1033 by source IP 216.35.217.56;
i. 1033, 1070 by source IP 216.35.217.51;
j. 1033, 1035, 1042, 1050, 1111 by source IP 216.33.199.112;
k. 1034 by source IP 216.33.199.112;
l. 1034, 1058 by source IP 216.35.217.51;
m. 1034, 1052, 1059, 1064, 1072 by source IP 216.33.199.91;
n. 1034, 1049, 1055, 1059, 1156, 1165, 1181, 1192 by source IP 216.33.199.19;
o. 1036, 1038, 1040, 1043, 1045 by source IP 216.33.199.119;
p. 1036, 1087, 1096, 1105, 1109, 1113, 1125, 1132, 1145, 1153, 1156, 1160 by source IP 216.35.217.53;
q. 1038, 1079, 1124 by source IP 216.32.73.120;
r. 1040 by source IP 216.33.210.64;
s. 1043 by source IP 216.33.210.69;
t. 1052, 1273, 1380, 1486, 1554 by source IP 216.33.199.118;
u. 1059, 1065, 1070, 1076 by source IP 216.33.199.113;
v. 1069 by source IP 216.33.210.68;
w. 1082, 1086, 1098, 1105, 1110, 1121, 1125, 1132, 1140, 1148, 1152, 1165, 1169, 1174 by source IP 216.33.199.114;
x. 1084 by source IP 216.33.199.94;
y. 1084, 1092, 1112 by source IP 216.35.217.52;
z. 2049 by source IP 24.114.58.156;
aa. 8888 by source IP 24.219.84.227;
bb. 20034 by source IP 24.2.95.205; and
cc. 65535 by source IP 24.13.185.141.

The port probes identified in subparagraph h-cc., above, in light green, with the exception of the last four shown, appear to originate from outside the *Cox@Home* domain. The IANA ports involved are well into the range of ephemeral ports typically shared with temporary client service requests, but no IANA assignments for these ports are known to exist. That these probes are specifically looking for ports in the range(s) reported by BID would imply that these ports, in addition to the ones seen in the TCP port scans, (above), have potential usefulness for hackers. If these ports are not assigned and are not in use, then they could be recruited as non-default trojan ports for modified or new exploits.

dd. 12436 (NetBus|GabanBus), 31337 (Back Orifice) by source IP 62.155.2246.154;
ee. **27347** (typo by a fumble-fingered hacker wannabe! – 27374 transposition error made) by source IP 208.61.109.243 ("it" also ran a SubSeven port probe at the same time - 3/24/00 12:40:51 - against port 12438 using the earlier version name "Sub 7"); and
ff. 27374 (Sub-7 2.1) by source IP 24.8.136.178, 24.3.69.120, 24.218.115.225, 195.240.205.40, 24.2.246.50, 24.94.206.148, 24.92.255.219, 24.226.111.97, 24.5.195.93, 171.211.66.120, 212.48.192.191, 24.6.136.16, 194.27.62.178, 207.192.77.192, 195. 229.254.170, 167.142.12.138, 24.27.173.107, 212.136.96.163, 24.4.89.56, 212.83.144.155, 24.6.199.157, 209.115.166.134, 24.15.117.68, 24.11.88.170, 24.11.88.2, 24.1.238.151(2), 63.226.134.240, 24.9.115.57, 63.226.134.240, 24.8.69.237(2), 24.8.69.68, 24.11.88.2, 24.11.88.170, 24.9.115.57. 24.130.60.93, 24.31.52.186, 24.114.209.149, 216.209.42.86, 207.172.239.122, 24.115.0.174, 207.172.55.93, 209.122.199.140, 24.6.183.103, 24.1.54.86, 62.6.100.45, 24.114.209.149, 24.240.72.64, 24.15.134.141, 24.8.254.186, 208.190.44.194, and 207.74.110.203.

The port probes identified in subparagraphs dd-ff., above, in red, appear to originate from both inside and outside the *Cox@Home* domain. The specific port numbers, like those shown in red for the TCP

port scans, are n=known trojan defaults. These probes are clearly hostile. One last observation is offered. The port bolded and underlines above, port 27347, is an obvious typographic error made by the hacker while crafting or customizing the attack packet! The detection immediately preceding this one used the same source IP address, 208.61.109.243, but was specifically a SubSeven port probe. Reusing the same source IP address is another indication this attacker is unsophisticated, but dangerous nevertheless.

Proxy Port probe: An intruder is scanning your system searching for a proxy server. The intruder then may be able to use your proxy server to browse the Internet anonymously. Currently, BID considers probes of TCP ports 3128, 8000, or 8080 to be proxy probes. Proxy probes seen appear to originate from both within as well as from outside the *Cox@Home* domain. Two of the ones seen to originate from outside are unidentified by the same source IP address (0.0.0.0) and are targeted against the well known port 8080, reserved by IANA for proxy servers. Another from outside originates from 204.162.66.206, and BID has determined that the name is "sun8.RTNA.DaimlerChrysler.COM." This site is the DaimlerChrysler Research and Technology Center in Palo Alto, CA. The "sun8" part of the name would imply that their SunMicrosystems server has likely been subverted and used without authorization to mask the true source of this hostile proxy probe against port 8080. Access to their web server confirmed this identity. Another proxy probe appeared to either originate from or be cloaked/disguised to be from a German bio-engineering firm or university, as the name returned by BID was "microbio7.biologie.uni-greifswald.de." An NT *tracert* tracked the name though the Cox network to the ATM switch in NYC, to Hannover/Hamburg, FRG, to Uni-Greifswald1.WiN-IP.DFN.DE [188.1.166.10], then "KR-Uni.Greifswald1.WiN-IP.DEN.DE [188.1.3.166], and finally to the name and IP source indicated. Attempts to access their web server to confirm this backtrace were unproductive. Another proxy probe attempted to elicit response from destination port 3128, in addition to 8080. A check against the list of IANA port assignments shows 3128 not to be a reserved port. The 192.232.248.198 source IP used for this proxy probe was bactraced by BID, but no name was able to de developed for further investigation. A subsequent instance of proxy port probing for 3128 as well as 8080 was later seen from source IP 213.172.2.204, and again, no backtrace by BID for the name used by the attacker was productive. Additional proxy probes from within the *Cox@Home* domain seemed to limit their adventures to port 8080, as did others from outside the domain. But nothing remarkable in them was noted for inclusion at this time.

DNS Port probe: This is a "Black Hat" attempt to determine if DNS service is available on your system. This can be done in preparation for a future attack, or to see if your system might be susceptible to attack. BID reports these as a false-positive detection if a DNS application is configured, but is temporarily unavailable.

BID detected 36 DNS port probes, of which 31 originated predominantly from outside the *Cox@Home* domain. The 5 source IPs seen starting within inside Cox were considered benign as there was nothing extraordinary about them. All other remaining source IPs from outside Cox were unexpected and in some case, very stealthy. These were all considered hostile, and included the following, some of which were repeated as indicated by the number in parentheses accompanying the IP:

- 210.220.213.251(2) and 209.216.2.200(2), with no backtrace name found by BID;
- 216.112.78.34 with the BID backtrace name "tsmtec.com";
- 216.102.128.180 with the BID backtrace name "adsl-216-102-128-180.dsl.lsan03.pacbell.net";
- 216.102.225.34 with the BID backtrace name "adsl-216-102-225-34.dsl.lsan03.pacbell.net";
- 63.201.114.36 with the BID backtrace name "radiomanila.net";
- 0.0.0.0**(24)** with no backtrace name found by BID – **24 instances from 4/3-4/11/00 is significant;**
- 199.111.112.120 with the BID backtrace name "xml1.nsu.edu";
- 213.8.219.30 with no backtrace name from BID.

The brace of unidentifiable DNS port probe requests from 0.0.0.0 reflect a very recent and intense period of hostile DNS probing on the Cox network, and this remains unexplained.

MSRPC Port probe: This is a "Black Hat" attempt to determine if RPC service is available on your system. This can be done in preparation for a future attack, or to see if your system might be

susceptible to attack. BID reports these as a false-positive detection if an RPC application is configured, but is temporarily unavailable.

BID detected only two MSRPC port probes, and both of these originated from outside the *Cox@Home* domain. Both of these detections were considered hostile, and include the following IP:

- 0.0.0.0(2) with no backtrace name found by BID.

As both unidentifiable MSRPC port probe requests came from 0.0.0.0, they reflect hostile probing on the Cox network that cannot be unexplained by any benign motive.

RPC Port probe: This is a "Black Hat" attempt to access Sun Remote Procedure Call (RPC) (rpcbind, portmapper) services on your system. It is probably caused by a sweep of millions of machines on the Internet that are Sun Microsystems product running Solaris or SunOS, or some other compatible Unix OSs that support RPC service. If you are not one of these, and do not run an implementation of this on your Win.x or WinNT host, then you are not at risk. RPC (Remote Procedure Call) is a networking technology developed by Sun Microsystems. It is used on most UNIX machines, and is a prevalent way to build networked applications that use remote tasks for distributed processing, a forerunner of the client-server model architecture.

Its continued popularity persists due to legacy enterprise systems that still run these services and are still vulnerable to the same security pitfalls they have always endured. When a hacker probes for RPC services, it is an early indication of a hostile intent. If an RPC service is found on your system, then the next hacker step that should be expected is an RPC portmapper dump. This will list all RPC programs on your machine and enable the hacker to determine the vulnerabilities of each. After checking hacker sites for published lists of known exploits, the hacker can then be expected to attempt to break into your system using one of the exploits found.

BID detected 20 RPC port probes, of which 11 of originated from outside the *Cox@Home* domain. The 9 source IPs seen starting within inside Cox were considered benign as there was nothing extraordinary about them. All other remaining source IPs from outside Cox were unexpected and in some case, very stealthy. These were all considered hostile, and included the following:

- 202.30.26.72 with the BID backtrace name "mach.ajou.ac.kr";
- 216.227.17.17 with the BID backtrace name "dsl-216-227-17-12.chi.interchangedsl.com";
- 208.49.251.3 with no backtrace name from BID;
- 131.104.204.156 with no backtrace name from BID;
- 210.96.22.193 with no backtrace name from BID;
- 203.230.240.86 with no backtrace name from BID;
- 210.109.56.32 with no backtrace name from BID;
- 129.219.245.39 with the BID backtrace name "noname-13633.generic.asu.edu";
- 63.248.50.194 with the BID backtrace name "63-248-50-194.usa3.flashcom.net";
- 203.127.42.155 with no backtrace name from BID;
- 208.232.120.196 with no backtrace name from BID.

The brace of 7 unidentifiable RPC port probe requests with no traceable name reflects hostile RPC probing on the Cox network, and this remains unexplained.

Telnet Port probe: The attacker uses this probe to scan your system to see if telnet service is configured and running. This program is enabled on most UNIX systems, but on virtually no Windows systems, so Win9.x and WinNT users are not at risk unless they have installed servers to do accept telnet connections for remote access. Telnet is a service that allows one machine to receive a command prompt, similar to a DOS prompt, which permits a remote access connection exchange on with the telnet server on another machine on the Internet. Windows comes with a telnet client that allows them to log into UNIX machines this way, but they do not ship with a built in server to enable them to host this service directly. This means that hackers cannot obtain a DOS prompt on your Windows machine unless you have also installed the special software necessary to run a telnet server.

Virtually all Unix machines have this service installed and running. Because there are so many telnet exploits that permit anyone (including a hacker) to login without authenticating (supplying a valid username and password), or to bypass or defeat these weak authentication measures, these systems suffer from a high risk of break-in. On Unix systems the hacker is almost certainly scanning millions of machines and checking for login banners that freely divulge useful information about themselves. Unix machines are probably secure from login attack, but other machines, such as routers and dial-up servers, often use telnet for remote management by System Administrator personnel. These devices are not always convenient to physically approach and access without the SysAdministrator having leave the console normally used to manage the rest of the enterprise. As dial-up services are frequently associated with telnet service use, hackers like to probe for telnet access to attack remote devices for either break-in or DoS attack purposes. False positives from BID for telnet probes when the service is enabled and actual service interruption occurs, making the device unavailable, so that the access attempt is interpreted as a detection instead.

BID detected 13 telnet port probes, of which 10 originated from outside the *Cox@Home* domain. The 8 source IPs seen starting within inside Cox were considered benign as there was nothing extraordinary about them. All other remaining source IPs from outside Cox were unexpected and in some case, very stealthy. These were all considered hostile, and included the following:

- 207.136.67.12 with the BID backtrace name "ministry.idirect.com;"
- 171.216.28.142 with the BID backtrace name "ABD81C8E.ipt.aol.com";
- 129.252.233.111 with no backtrace name from BID;
- 206.109.121.111 with the BID backtrace name "s82.max2.Galveston.box.net";
- 210.55.82.149 with the BID backtrace name "210-55-82-149.dialup.xtra.co.nz";
- 145.236.217.153 with the BID backtrace name "line-217-153.dial.matav.net";
- 212.204.137.53 with the BID backtrace name "cc13991-a.zwoll1.ov.nl.home.com";
- 198.164.140.85 with no backtrace name from BID;
- 203.230.240.86 with no backtrace name from BID;
- 63.225.106.153 with no backtrace name from BID;
- 216.221.109.158 with the BID backtrace name "tc-920.dialup.srtnet.com";
- 209.167.106.20 with no backtrace name from BID;
- 62.158.190.225 with the BID backtrace name "p3E9EBEE1.dip0.t-ipconnect.de".

The brace of unidentifiable RPC port probe requests with no traceable name reflects hostile RPC probing on the Cox network, and this remains unexplained.

Summary of Trojan Probes: Trojan horse program probes are an indication that someone has unsuccessfully attempted to find default communication ports to any of several potential backdoors into your computer. When no response from your system is forthcoming, the attacker goes away, and you system remains operating without being compromised. Some trojan programs use UDP ports as well as TCP ports, as detailed at *www.doshelp.com/trojanports.htm*, but none of these were seen during this 60 day monitoring period.

A trojan program is one that has some subversive purpose other than what it purports to support. One of the favorite "Black Hat" techniques is to send these programs to people in email, news programs, or chat rooms in the hope that they will be duped into running them. Once run, they may be come thereafter undetectable except for the activity (communications and processing) they will exhibit when operated by the attacker. Typical trojans are those that process remote commands from outside your computer, steal userids and passwords, install viruses, erase log files, reformat your hard-disk, and other malicious acts, all without your direct knowledge. As indicated, one particular popular class of trojan programs are the Remote Access Trojans. These are programs that provide the hacker complete remote control over your machine, despite the fact that the attacker doesn't know where on the Internet your computer is located. When this becomes necessary for them to do further damage, for example, they must scan the entire range of your ISP addresses to try to find you. Trojan probes are done to identify computers that have been "trojanized," either by the "Black Hat" running the probe, or by another hacker that has gotten into one implanted. They are not really picky and will accept and run other hackers Trojans besides their own.

Trojan Horse probes are very common, growing in sophistication with mutation and updates. Many are identifiable by anti-virus programs and can be detected and removed from email, thus greatly reducing the threat that this delivery method has as a transmission vector.

NetBus probe: Netbus is a Win32 based remote access trojan (RAT) program. This trojan can affect Win9.x and WinNT OSs. Netbus trojans must be received and executed for self-installation as a precondition to "Black Hat" use. As part of the self-install, it updates the Windows Registry so that it remains active all the time. The current version of Netbus seen is v2.0, which the author is attempting to sell as a legitimate network management tool. Three earlier versions of Netbus exist, besides v2.1:

- Netbus v1.5 is 473,088 bytes – released in March '98;
- Netbus v1.6 is 472,576 bytes – released in August '98; and
- Netbus v1.7 is 494,592 bytes – released in November '98.

Netbus is a remote administration trojan server program similar to BackOrifice. While you are connected to the Internet, if this server program is activated, anyone that executes the Netbus client program can connect to your computer through this covert channel, without your permission or knowledge, from anywhere at anytime. All information within your computer, including userid and passwords to databases and other networks, stored credit card numbers, resumes, email, and proprietary data files can be stolen. In addition, other unauthorized software can be installed and executed, your CD-ROM and mouse controlled, and all keystrokes copied and communicated through the same covert channel back to the Netbus client. If this Trojan is running on your host, your host is no longer under your control. Detailed information about the usage of Netbus server ports 12345 and 12346 are described at: http://www.nwi.net/~pchelp/nb/netbus.htm.

If your machine has been "trojanized" (infected and the infection activated), then it has been subverted. Use of a virus-protection software to remove this and other Trojans is the only cure, short of trying to manually remove the files and update the Windows Registry, a potentially tricky operation.

The latest Netbus v2.1 features include:

- Real Time Chat with server operator from the remote client admionistrator;
- Telnet access to host MS-DOS-prompt;
- HTTP access to host files, including download and upload using web-browser services;Host list integration with network neighborhood.
- Server setup and administration (close server, restrict IP access, TCP-port, password, visibility, access mode, auto start);
- General system information and cached passwords;
- Message manager;
- Window manager (full control over all windows);
- Registry manager (list keys, fields and values, create keys and delete keys, change values among others);
- Sound system (raise and lower volume);
- Plugin manager (run Plugins that extend the capabilities of NetBus);
- Port redirect (simple proxy support);
- Server host application redirection (e.g. allows remote interaction with MS-DOS prompts);
- Server file actions (execute executable files, show image files, play audio files, open document files and print document files);
- Server host spy functions (includes listen keyboard, get screen capture, record audio from microphone and get web camera image);
- Server host file manager (explorer, upload and download files, delete files and folders, create folders and share folders);
- Exit Windows (reboot system, shutdown system or power down system);
- Other functions (Client chat, open and close CD-ROM, disable keys, key click, swap mouse buttons, Go to URL, Send text);
- NetBus scanner, fast port scanner;

- Server host scheduler, predefine time to run scripts at hosts;
- Client command broadcaster, broadcasts commands to multiple server hosts;
- Multi-language support, extendable to more languages than just English.;
- Skin support (transparent backgrounds); and
- Install Wizard and Online help manual.

BID detected 4 Netbus port probes, of which 2 originated from outside the *Cox@Home* domain, and 2 from within. All 4 were backtraced using tracert and the findings are presented below. All 4 were unexpected and considered hostile, and detailed below:

- 216.112.169.111 with the BID backtrace name "ts008d03.par-nj.concentric.net";
- 62.155.246.154 with the BID backtrace name "p3E9BF69A.dip0.t-ipconnect.de";
- 24.2.95.205 with the BID backtrace name "cc940951-a.stcl1.mi.home.com";
- 24.108.105.6 with the BID backtrace name "cd852627-a.ctjams1.mb.wave.home.com"

.

The 216.112.169.111 address was *tracert*'ed back to a Paramus, NJ customer of the Concentric Network Corporation, *http://www.concentric.com/corporate_info/index.html*, a San Jose, CA based Internet provider, whose services include high-speed dedicated and DSL access.

The 62.155.246.154 address was *tracert*'ed back to a German customer of the Eris Free Network, the self-proclaimed "first and largest IRC network , eris being eris.berkelet.edu." This organization has been in existence since Aug 1990. The website, *http://www.efnet.net/about.html*, claims its channel is used for conversations in German, Finnish, Russian, Japanese, and occasionally other languages.

The 24.2.95.205 address was *tracert*'ed back to a St.Clair, MI customer of the ExciteHome.Net, and after consulting *http://www.networksolutions.com/cgi-bin/whois/whois?STRING=home.com* and *www.mapquest.com* and doing a search within Michigan for a city with the beginning initials "stcl", it was discovered that St.Clair, MI was the likely location where the attack originated, but no further information was obtained which would reveal something further about the attacker.

The 24.108.105.6 address was *tracert*'ed back through: Phoenix, San Diego, Tulsa, Omaha, Des Moines, Chicago, Cleveland, and Buffalo "bb2-pos5-0.rdc1.on.home.net [24.7.74.26] and IPs 10.0.185.22, 10.0.185.26, 10.0.185.18, and finally 10.0.185.10 before timing out. Repeated attempts to complete this trace timed out at after approximately the same duration. After once again consulting the WHOIS directory, *http://www.networksolutions.com/cgi-bin/whois/whois?STRING=home.com* and *www.mapquest.com*, no further information was obtained which would reveal something further about the attacker. But, it is nevertheless speculated that the attack originated from a city with the initials "ctjams", possibly someplace named Court James, in Manotiba, Canada, as the city portion of the DNS bactrace returned by BID revealed the initials "mb".

SubSeven probe: This is one of most favored "Black Hat" trojans seen and reported on the Internet. Versions 2.1, 2.0, 1.9 and older versions are currently in the wild. Detections made and reported by BID during this 60 day period include those of Sub7 (version non-specific) and Sub7_2. Version 2.2 beta release has been announced and should be expected to be appear in trace logs in the near future. Features of each SubSeven version release, as well as those of EditServer, are documented online at *http://subseven.slak.org/features.html*, and a 4/13/00 post indicates an intent to open a new "SKINS" page, presumably for transparent backgrounds. GUI screens shots of the SubSeven v2.1 M.U.I.E. are also posted by mobman, the code author and webmaster.

- Features added in 2.1:
  - address book
  - WWP Pager Retriever
  - UIN2IP
  - remote IP scanner
  - host lookup
  - get Windows CD-KEY

update victim from URL
ICQ takeover
FTP root folder
retrieve dial-up passwords along with phone numbers and usernames
port redirect
IRC bot. for a list of commands, click here
File Manager bookmarks
make folder, delete folder [empty or full]
process manager
text 2 speech
clipboard managerEDITSERVER CHANGES

- EditServer for 2.1 changes:
  customizable colors
  change server ICON
  pick random port on server startup
  irc bot configuration

- Features added in 2.0:
  Restart server
  Aol Instant Messenger Spy
  Yahoo Messenger Spy
  Microsoft Messenger Spy
  Retrieve list of ICQ users and passwords
  Retrieve list of AIM users and passwords
  App Redirect
  Edit file
  Perform clicks on victim's desktop
  Set/Change Screen Saver settings [Scrolling Marquee]
  Restart Windows [see below]
  Ping server
  Compress/Decompress files before and after transfers
  The Matrix
  Ultra Fast IP scanner [thanks to Blade's TH]
  IP Tool [Resolve Host names/Ping IP addresses]
  Get victim's home info [not possible on all servers]:

  - Address
  - Business name
  - City
  - Company
  - Country
  - Customer type
  - E-Mail
  - Real name
  - State
  - City code
  - Country code
  - Local Phone
  - Zip code

  Configure Client colors
  Configure menu options [add/delete pages, change names]
  Automatically Display Image when downloaded [jpg,bmp]
  Automatically edit files when downloaded [txt,bat]
  Change port numbers for The Matrix, Keylogger and Spies

Retrieve "SubSeven message of the day"

- EditServer for 2.0 new features:
  Protect server's Port and Password once installed
  Melt server when executed
  Protect server settings with a password

- 1.9 or older features:
  Open Web Browser to specified location.
  Restart Windows [5 methods]:

  - Normal shutdown
  - Forced Windows shutdown
  - Log off Windows user
  - Shutdown Windows and turn off computer
  - Reboot System

  Reverse/restore Mouse buttons.
  Hide/Show Mouse Pointer.
  Control Mouse.
  Mouse Trail Configuration.
  Set Volume.
  Record Sound file from remote microphone.
  Change Windows Colors / Restore.
  Hung up Internet Connection.
  Change Time.
  Change Date.
  Change Screen resolution.
  Hide Desktop Icons / Show
  Hide Start Button / Show
  Hide taskbar / Show
  Open CD-ROM Drive / Close
  Beep computer Speaker / Stop
  Turn Monitor Off / On
  Disable CTRL+ALT+DEL / Enable
  Turn on Scroll Lock / Off
  Turn on Caps Lock / Off
  Turn on Num Lock / Off
  Connect / Disconnect
  Fast IP Scanner
  Get Computer Name
  Get User Name
  Get Windows and System Folder Names
  Get Computer Company
  Get Windows Version
  Get Windows Platform
  Get Current Resolution
  Get DirectX Version
  Get Current Bytes per Pixel settings
  Get CPU Vendor
  Get CPU Speed
  Get Hard Drive Size
  Get Hard Drive Free Space
  Change Server Port
  Set/Remove Server Password
  Update Server

Close Server
Remove Server
ICQ Pager Connection Notify
IRC Connection Notify
E-Mail Connection Notify
Enable Key Logger / Disable
Clear the Key Logger Windows
Collect Keys pressed while Offline
Open Chat Victim + Controller
Open Chat among all connected Controllers
Windows Pop-up Message Manager
Disable Keyboard
Send Keys to a remote Window
ICQ Spy
Full Screen Capture
Continues Thumbnail Capture
Flip Screen
Open FTP Server
Find Files
Capture from Computer Camera
List Recorded Passwords
List Cached Passwords
Clear Password List
Registry Editor
Send Text to Printer
Show files/folders and navigate
List Drives
Execute Application
Enter Manual Command
Type path Manually
Download files
Upload files
Get File Size
Delete File
Play *.WAV
Set Wallpaper
Print .TXT\.RTF file
Show Image
List visible windows
List All Active Applications
Focus on Window
Close Window
Disable X (close) button
Hide a Window from view.
Show a Hidden Window
Disable Window
Enable Disabled Window
Set Quality of Full Screen Capture
Set Quality of Thumbnail Capture
Set Chat font size and Colors
Set Client's User Name
Set local 'Download' Directory
Set Quick Help [Hints]

- EditServer for 1.9 or older features:
  PreSet Target Port

PreSet server Password
Attach EXE File
PreSet filename after installation
PreSet Registry Key
PreSet Autostart Methods:

- Registry: Run
- Registry: RunSevices
- Win.ini
- Less known method
- Not known method

PreSet Fake error message
PreSet Connection Notify Username
PreSet Connection Notify to ICQ#
PreSet Connection Notify to E-Mail
PreSet Connection Notify to IRC Channel or nickname port redirect
IRC commands
File Manager bookmarks
Make folder, delete folder [empty or full]
Process manager
Text 2 speech
Clipboard managerEDITSERVER CHANGES

BID detected 21 SubSeven port probes, of which 9 originated from outside the *Cox@Home* domain, and 12 from within. All 9 from outside were backtraced using *tracert* and the findings of the first 4 of these are presented below with the actual traces as completed, for additional detail. The ones originating from within the *Cox@Home* domain were not *tracert*'ed as these have already been established to be unproductive from earlier efforts on other attack signatures. However, all 21 are unexpected and considered hostile, and detailed below:

- 24.8.213.116 with the BID backtrace name "c498234-a.donora1.pa.home.com";
- 24.2.246.50 with the BID backtrace name "c495697-a.ankeny1.ia.home.com";
- 24.129.8.203 with the BID backtrace name "dsf-8-203.jacksonville.net";
- 24.42.84.63 with the BID backtrace name "cr322417-a.flfrd1.on.wave.home.com";
- 24.67.127.237 with the BID backtrace name "24.67.127.237.ab.wave.home.com";
- 62.155.246.154 with the BID backtrace "p3E9BF69A.dip0.t-ipconnect.de";
- 208.61.109.71 with the BID backtrace name "adsl-61-109-71.sdf.bellsouth.net";
- 209.214.168.67 with the BID backtrace name "host-209-214-168-67.sdf.bellsouth.net";
- 166.62.2.240 with the BID backtrace name "usr4-dialup48.mix1.Sacramento.cw.net";
- 24.48.24.143 with the BID backtrace name "surf15-24-143.dad.adelphia.net";
- 24.112.50.10 with the BID backtrace name "cr468063-a.lndn1.on.wave.home.com";
- 24.112.202.116 with the BID backtrace name "cr143622-a.cambr1.on.wave.home.com";
- 24.112.50.21 with the BID backtrace name "cr183282-b.lndn1.on.wave.home.com";
- 213.1.87.205 with no backtrace name from BID;
- 208.61.109.243 with the BID backtrace name "adsl-61-109-243.sdf.bellsouth.net"
- 24.114.29.198 with the BID backtrace name "cr83063-c.pr1.on.wave.home.com";
- 216.254.154.13 with the BID backtrace name "dialin-154-13.tor.primus.ca";
- 216.209.209.234 with the BID backtrace name "HSE-Montreal-ppp101429.simpatico.ca";
- 128.230.221.80 with the BID backtrace name "syru221-080.syr.edu";
- 24.27.184.182 with the BID backtrace name "cvg-27-184-182.cinci.rr.com";
- 24.29.26.152 with the BID backtrace name "cvg-026-152.cinci.rr.com".

The 62.155.246.154 address was *tracert*'ed back to a German customer of the Eris Free Network, the self-proclaimed "first and largest IRC network , eris being eris.berkelet.edu." This organization has

been in existence since Aug 1990. The website, *http://www.efnet.net/about.html*, claims its channel is used for conversations in German, Finnish, Russian, Japanese, and occasionally other languages.

The 208.61.109.71 address was *tracert*'ed back to a USA customer of the Bellsouth Network, in the Louisville, KY area as the likely location where the attack originated, but no further information was obtained which would reveal something further about the attacker. The trace passed back through: Phoenix, San Diego, Anaheim, then to the San Jose's ATM switch, and the St. Louis ATM switch, and Louisville, KY gateway, "bs-lousiville-gw.customer.ALTER.NET [157.130.97.250] and IPs 205.152.133.134 and finally 205.152.133.74 before timing out. Repeated attempts to complete this trace timed out at after approximately the same duration. After once again consulting the WHOIS directory, *http://www.networksolutions.com/cgi-bin/whois/whois?STRING=bellsouth.net* and *www.mapquest.com*, no further information was obtained which would reveal something further about the attacker. But, it is felt that the attack originated from a city with the initials "sdf" as these seem to be the encoding schemes used, but slightly varying, amongst Internet providers. Repeated attempts to complete this trace timed out at after approximately the same duration. After once again consulting the WHOIS directory, *http://www.networksolutions.com/cgi-bin/whois/whois?STRING=bellsouth.net* and *www.mapquest.com*, no further information was obtained which would reveal something further about the attacker. But, it is felt that the attack originated from a city with the initials "sdf" as these seem to be the encoding schemes used, but slightly varying, amongst Internet providers.

The 209.214.169.67 address was *tracert*'ed back to a USA customer of the Bellsouth Network, in the Louisville, KY area as the likely location where the attack originated, but no further information was obtained which would reveal something further about the attacker. The trace passed back through: Phoenix, San Diego, Anaheim, then to the San Jose's ATM switch, and the St. Louis ATM switch, and Louisville, KY gateway, "bs-lousiville-gw.customer.ALTER.NET [157.130.97.250] and IPs 205.152.133.159 and finally 209.215.214.129 before successfully completing. It is felt that the attack originated from a city with the initials "sdf" as these seem to be the encoding schemes used, but slightly varying, amongst Internet providers. A copy of the trace is provided below:

```
Tracing route to host-209-214-168-67.sdf.bellsouth.net [209.214.168.67]
over a maximum of 30 hops:

  1    50 ms    70 ms   <10 ms  r1-fe0-0-100bt.chnd1.az.home.net [24.1.208.1]
  2    10 ms    20 ms   <10 ms  10.0.208.1
  3   <10 ms    10 ms    10 ms  c1-pos5-0.phnxaz1.home.net [24.7.72.5]
  4    30 ms    10 ms    20 ms  c1-pos7-0.sndgca1.home.net [24.7.65.134]
  5    10 ms    10 ms    60 ms  c1-pos1-0.anhmca1.home.net [24.7.64.69]
  6    40 ms    50 ms    40 ms  24.7.65.165
  7   180 ms   180 ms   171 ms  148.ATM1-0.BR2.sjc1.ALTER.NET [137.39.91.25]
  8   190 ms   170 ms     *     154.ATM3-0.XR2.SJC1.ALTER.NET [152.63.51.178]
  9   190 ms   191 ms   210 ms  193.at-1-0-0.TR4.SCL1.ALTER.NET [152.63.48.250]
 10   261 ms   270 ms   250 ms  207.ATM6-0.TR2.STL1.ALTER.NET [152.63.3.241]
 11   281 ms   270 ms   291 ms  296.ATM7-0.XR2.STL1.ALTER.NET [146.188.224.61]
 12   280 ms   270 ms   271 ms  192.ATM9-0-0.GW1.STL1.ALTER.NET [146.188.224.77]
 13   290 ms   391 ms   290 ms  bs-louisville-gw.customer.ALTER.NET [157.130.97.250]
```

```
 14   261 ms   260 ms   250 ms  205.152.133.159

 15     *      290 ms   280 ms  209.215.214.129

 16   571 ms   541 ms   440 ms  host-209-214-168-67.sdf.bellsouth.net [209.214.168.67]


Trace complete.
```

The 166.62.2.240 address was *tracert*'ed back to a USA dialup customer of the Sacramento Network, in the Sacramento, CA area as the likely location where the attack originated, but no further information was obtained which would reveal something further about the attacker. The trace passed back through: Phoenix, San Diego, Anaheim, then to the San Francisco re-router and fiber network (fddio) where the destination host was found to be unreachable. The trace terminated normally. A copy of the trace is provided below:

```
Tracing route to usr4-dialup48.mix1.Sacramento.cw.net [166.62.2.240]

over a maximum of 30 hops:


  1    10 ms   <10 ms    10 ms  r1-fe0-0-100bt.chnd1.az.home.net [24.1.208.1]

  2    10 ms   <10 ms   <10 ms  10.0.208.1

  3    10 ms   <10 ms    10 ms  c1-pos5-0.phnxaz1.home.net [24.7.72.5]

  4    20 ms    10 ms    10 ms  c1-pos7-0.sndgca1.home.net [24.7.65.134]

  5    10 ms    20 ms    10 ms  c1-pos1-0.anhmca1.home.net [24.7.64.69]

  6    40 ms    50 ms    30 ms  24.7.65.165

  7    60 ms    50 ms    50 ms  206.24.241.9

  8    40 ms    40 ms    40 ms  acr2-loopback.SanFranciscosfd.cw.net [206.24.210.62]

  9    20 ms    30 ms    40 ms  corerouter2.SanFrancisco.cw.net [204.70.9.132]

 10    30 ms    30 ms    31 ms  core1.Sacramento.cw.net [204.70.4.225]

 11    30 ms    40 ms    50 ms  mix1-fddi0-0.Sacramento.cw.net [204.70.164.43]

 12  usr4.mix1.Sacramento.cw.net [166.62.0.41]  reports: Destination host unreachable.
```

Trace complete.

The 213.1.87.205 address was *tracert*'ed back to a London customer of the BT Internet Network, at: http://www.btinternet.com/index.html. This organization is based in London and is a member of the Internet Watch Foundation, a group dedicated to fighting child pornography. No further information was obtained which would reveal something further about the attacker. It is very curious that the IP address trace completed for this one instance of attacks signatures seen during the 60 day period was found to also route through the San Francisco corerouter, as did the other one examined just above [162.62.2.240]. A copy of the trace is provided below:

```
Tracing route to host213-1-87-205.btinternet.com [213.1.87.205]

over a maximum of 30 hops:
```

```
  1   <10 ms   <10 ms    10 ms  r1-fe0-0-100bt.chnd1.az.home.net [24.1.208.1]
  2   <10 ms    10 ms   <10 ms  10.0.208.1
  3    10 ms    10 ms    10 ms  c1-pos5-0.phnxaz1.home.net [24.7.72.5]
  4    10 ms    10 ms    10 ms  c1-pos7-0.sndgca1.home.net [24.7.65.134]
  5    20 ms    10 ms    20 ms  c1-pos1-0.anhmca1.home.net [24.7.64.69]
  6    20 ms    20 ms    20 ms  24.7.65.165
  7    20 ms    30 ms    20 ms  206.24.241.9
  8    20 ms    20 ms    30 ms  acr2-loopback.SanFranciscosfd.cw.net [206.24.210.62]
  9    30 ms    50 ms    20 ms  corerouter2.SanFrancisco.cw.net [204.70.9.132]
 10    90 ms    90 ms   101 ms  corerouter2.WestOrange.cw.net [204.70.9.139]
 11   100 ms   100 ms   110 ms  acr2-loopback.NewYorknyr.cw.net [206.24.194.62]
 12   390 ms   411 ms   400 ms  bcr1-so-1-0-0.London.cw.net [166.63.163.46]
 13   611 ms   591 ms   601 ms  bt-telecommunications.London.cw.net [166.63.163.170]
 14   601 ms   601 ms   611 ms  194.72.9.176
 15   601 ms   601 ms   621 ms  imsnte5.bt.net [212.140.207.146]
 16   601 ms   591 ms   591 ms  inh1r1-gigabit1.ims.bt.net [193.113.185.218]
 17   601 ms   601 ms   601 ms  192.168.251.23
 18   601 ms   591 ms   601 ms  192.168.255.4
 19   721 ms   721 ms   971 ms  host213-1-87-205.btinternet.com [213.1.87.205]

    Trace complete.
```

The 208.61.109.243 address was *tracert*'ed back to a USA customer of the Bellsouth Network, in the Louisville, KY area as the likely location where the attack originated, but no further information was obtained which would reveal something further about the attacker. The trace passed back through: Phoenix, San Diego, Anaheim, then to the San Jose's ATM switch, and the St. Louis ATM switch, and Louisville, KY gateway, "bs-lousiville-gw.customer.ALTER.NET [157.130.97.250] and IPs 205.152.133.134 and finally 205.152.133.74 before timing out. Repeated attempts to complete this trace timed out at after approximately the same duration. After once again consulting the WHOIS directory, *http://www.networksolutions.com/cgi-bin/whois/whois?STRING=bellsouth.net* and *www.mapquest.com*, no further information was obtained which would reveal something further about the attacker. But, it is felt that the attack originated from a city with the initials "sdf" as these seem to be the encoding schemes used, but slightly varying, amongst Internet providers.

The 216.254.154.13 address was *tracert*'ed back to a Canadian customer of the Primus Network, at: *http://www.primus.ca*. As can be seen from the trace, as from all the ones preceding this one, the path taken back to the subnet domain of origin eventually timed out, and no further information about the attacker was established. A copy of the trace is provided below:

```
Tracing route to dialin-154-13.tor.primus.ca [216.254.154.13]
```

```
over a maximum of 30 hops:

  1   <10 ms    10 ms   <10 ms  r1-fe0-0-100bt.chnd1.az.home.net [24.1.208.1]
  2   <10 ms   <10 ms    10 ms  10.0.208.1
  3   <10 ms   <10 ms    10 ms  c1-pos5-2.phnxaz1.home.net [24.7.74.5]
  4    10 ms    20 ms    10 ms  c1-pos7-0.sndgca1.home.net [24.7.65.134]
  5    30 ms    10 ms    20 ms  c1-pos1-0.anhmca1.home.net [24.7.64.69]
  6    20 ms    20 ms    30 ms  24.7.65.165
  7   180 ms   170 ms   160 ms  148.ATM1-0.BR2.sjc1.ALTER.NET [137.39.91.25]
  8   160 ms   170 ms   170 ms  154.ATM2-0.XR2.SJC1.ALTER.NET [152.63.51.186]
  9   180 ms   170 ms   160 ms  192.at-2-1-0.TR2.SAC1.ALTER.NET [152.63.51.54]
 10   240 ms   240 ms   241 ms  127.at-6-1-0.TR2.NYC9.ALTER.NET [152.63.6.81]
 11   250 ms   260 ms   251 ms  186.ATM6-0.XR2.NYC4.ALTER.NET [152.63.21.137]
 12   220 ms   241 ms   230 ms  188.ATM9-0-0.GW5.NYC4.ALTER.NET [146.188.179.229]
 13   260 ms   291 ms   280 ms  224.ATM1-0-0.BB1.TOR2.UUNET.CA.ALTER.NET [137.39.75.26]
 14   291 ms   290 ms   290 ms  f0-0-0.bb2.tor2.uunet.ca [205.150.242.110]
 15   311 ms   320 ms   291 ms  e-020.gw-1.tor.accglobal.net [205.150.89.10]
 16   290 ms   281 ms   290 ms  fe-0-0-0.rommel.tor.primus.ca [216.254.128.130]
 17     *        *        *     Request timed out.


    Trace complete.
```

The 216.209.209.234 address was *tracert*'ed back to a Canadian customer of the Sympatico Network, at: *http://www.sympatico.ca/*. As can be seen from the trace, as from all the ones preceding this one, the path taken back to the subnet domain of origin eventually timed out with a maximum of 30 hops, and no further information about the attacker was established. A copy of the trace is provided below:

```
Tracing route to 216.209.209.234 over a maximum of 30 hops

  1     *        *        *     Request timed out.
  2     *        *        *     Request timed out.
  3     *        *        *     Request timed out.
  4     *        *        *     Request timed out.
  5     *        *        *     Request timed out.
  6     *        *        *     Request timed out.
  7     *        *        *     Request timed out.
```

```
 8     *        *        *     Request timed out.
 9     *        *        *     Request timed out.
10     *        *        *     Request timed out.
11     *        *        *     Request timed out.
12     *        *        *     Request timed out.
13     *        *        *     Request timed out.
14     *        *        *     Request timed out.
15     *        *        *     Request timed out.
16     *        *        *     Request timed out.
17     *        *        *     Request timed out.
18     *        *        *     Request timed out.
19     *        *        *     Request timed out.
20     *        *        *     Request timed out.
21     *        *        *     Request timed out.
22     *        *        *     Request timed out.
23     *        *        *     Request timed out.
24     *        *        *     Request timed out.
25     *        *        *     Request timed out.
26     *        *        *     Request timed out.
27     *        *        *     Request timed out.
28     *        *        *     Request timed out.
29     *        *        *     Request timed out.
30     *        *        *     Request timed out.

Trace complete.
```

The 128.230.221.80 address was *tracert*'ed back to a user of the Syracuse University, Syracuse, NY, at: *http://www.syr.edu*. As can be seen from the trace, the path taken back to the subnet domain of origin was completed and the source there verified. A copy of the trace is provided below:

```
Tracing route to syru221-080.syr.edu [128.230.221.80]

over a maximum of 30 hops:


  1    10 ms   <10 ms   <10 ms  r1-fe0-0-100bt.chnd1.az.home.net [24.1.208.1]
  2   <10 ms   <10 ms    10 ms  10.0.208.1
  3    10 ms   <10 ms    10 ms  c1-pos5-0.phnxaz1.home.net [24.7.72.5]
```

```
  4    10 ms    10 ms    20 ms  c1-pos7-0.sndgca1.home.net [24.7.65.134]

  5    20 ms    20 ms    40 ms  c1-pos1-0.anhmca1.home.net [24.7.64.69]

  6    20 ms    30 ms    20 ms  24.7.65.165

  7    30 ms    20 ms    20 ms  sl-gw8-sj-0-3.sprintlink.net [144.232.192.157]

  8    20 ms    30 ms    20 ms  sl-bb11-sj-7-0.sprintlink.net [144.232.3.113]

  9    20 ms    20 ms    20 ms  sl-bb10-sj-9-0.sprintlink.net [144.232.3.25]

 10    70 ms    70 ms    70 ms  sl-bb12-rly-5-0.sprintlink.net [144.232.9.217]

 11    70 ms    70 ms    70 ms  sl-gw9-rly-9-0.sprintlink.net [144.232.7.254]

 12   111 ms    90 ms    80 ms  at-gw4-syr-1-0-0-OC3.appliedtheory.net [169.130.1.141]

 13    90 ms    90 ms    90 ms  at-gw5-syr-4-0-0-OC3.appliedtheory.net [169.130.2.98]

 14   100 ms   100 ms   110 ms  169.130.33.10

 15   100 ms   100 ms    90 ms  128.230.146.2

 16   120 ms    91 ms   110 ms  syr0-0100.syr.edu [128.230.165.1]

 17   100 ms   110 ms   101 ms  syru221-080.syr.edu [128.230.221.80]

    Trace complete.
    ------------------------------------------------------------------------------------------------------------------------
```

NetSphere probe: NetSphere v1.31337.zip is the final release of this Win9.x trojan program. The author is apparently under parental controls to cease and desist activity of this nature, as indicated at: *http://www.xploiter.com/security/netsphere.shtml*. Keystroke log functions were disabled, but ICQ functions were improved. The following components are found within this release:

- NetSphereClient.exe size is 426k;
- NetSphereServer.exe size is 281k; and
- ICSMAPI.dll size is 57k.

The server listens to default TCP ports 30100, 30101, 30102 & 30103 and UDP port 30103. The usage of TSP 30103 and UDP 30103 is not documented at *www.doshelp.com/trojanports.htm*. NetSphere is written in Delphi 4, is of Medium Risk, and affords a nice GUI, but lacks functions seen in Netbus and Back Orifice. If activated, it attempts to locate and send the following information to the NetSphere client:

- Windows Version
- Registered Owner
- Registered Organization
- Windows Directory
- Login/Profile Name
- Computer Name
- Workgroup Name
- Computer Description
- Processor
- Processor Type
- Address
- City
- Province/State
- Country
- Postal/Zip Code
- Phone Number
- Time Zone
- Total RAM

- Used RAM
-

Other features will permit the NetSphere client to:
- Turn on and off the server monitor;
- ICQ:
  - Add UIN to target's contact list;
  - Add Author to target's contact list;
  - Add target to your own contact list;
- Capture Screen;
- Mouse Control;
- File system;
- Application list is easy to manage, unlike the one in NetBus Pro. With each, you can:
  - Switch to
  - Terminate
  - Hide
  - Show
  - Minimize
  - Maximize
  - Restore
  - Change Caption
- You also get the following information about each window:
  - Caption
  - Executable
  - State (Invisible, Active Wnd, Active App)
- For multimedia you can record from 5 to 60 seconds of audio.
- Permits client batch file creation for invisible remote execution on the target server for:
  - Client Chat
  - Messaging
  - Internet Browsing
  - Shutdown
    - Shutdown and logoff, or
    - Hang their Pentium CPU.
- Show Image
  - Execute tools - Ping client, DNS Lookup (straight and reverse) client, and an IP scanner.
  -

Earlier versions of NetSphere are identified as follows:

- NetSphere version 1.27a (727kb) and contains the following files:
  - NetSphere.txt;
  - NetSphereClient.exe size is 1.01M – released 15 April 1999;
  - NetSphereServer.exe size is 621K – released 15 April 1999.
- NetSphere version 1.29 adds keystroke logging functions and contains the following files:
  - NetSphereClient.exe size is 1.05M - 17 May 1999;
  - NetSphereServer.exe size is 655K – released 17 May 1999; and
  - khd2.dll size is 41K – released 13 April 1999.

BID detected two NetSphere probes. The first consisted of a series of 55 TCP port probes all targeted against the NetSphere Trojan, attempting to activate it on the numerous ports indicated. The attack originated from outside the *Cox@Home* domain, but was backtraced using tracert. The findings of that and the second probe attempt are presented below. It was unexpected, fast, and is considered hostile, especially considering that first one went after 55 ports is a systematic and quick probe attack, with little fanfare and noise. The second was attributed to a return visit from the HSE-Montreal Point-To-Point dialer seen earlier in the analysis. It was not retraced as the other trace was inconclusive and unproductive:

- 63.224.159.228 with the BID backtrace name "63-224-159-228.customers.uswest.net";
- 216.209.209.234 with the BID backtrace name "HSE-Montreal-ppp.simpatico.ca".

The 63.224.159.228 address was *tracert*'ed back to a USA customer of US West, at: http://www.uswest.com, a Phoenix, AZ based Internet provider, whose services include high-speed dedicated and DSL access. As can be seen from the trace, the path taken back to the subnet domain of origin was completed and the source there verified. A copy of the trace is provided below:

```
Tracing route to 63-224-159-228.customers.uswest.net [63.224.159.228]

over a maximum of 30 hops:


  1    40 ms    10 ms   <10 ms  r1-fe0-0-100bt.chnd1.az.home.net [24.1.208.1]

  2   <10 ms   <10 ms    10 ms  10.0.208.1

  3    10 ms   <10 ms    10 ms  c1-pos5-0.phnxaz1.home.net [24.7.72.5]

  4    10 ms    20 ms    10 ms  c1-pos7-0.sndgca1.home.net [24.7.65.134]

  5    10 ms    10 ms    20 ms  c1-pos1-0.anhmca1.home.net [24.7.64.69]

  6    20 ms    20 ms    20 ms  24.7.65.165

  7   250 ms   250 ms   241 ms  148.ATM1-0.BR2.sjc1.ALTER.NET [137.39.91.25]

  8   240 ms   231 ms   240 ms  154.ATM3-0.XR2.SJC1.ALTER.NET [152.63.51.178]

  9   270 ms   260 ms   261 ms  192.at-1-1-0.TR2.SAC1.ALTER.NET [152.63.51.46]

 10   251 ms   250 ms   260 ms  127.at-6-1-0.TR2.LAX9.ALTER.NET [152.63.5.145]

 11   241 ms   250 ms   250 ms  296.ATM7-0.XR2.LAX2.ALTER.NET [152.63.112.165]

 12   260 ms   270 ms   271 ms  194.ATM9-0-0.GW1.PHX1.ALTER.NET [146.188.249.125]

 13   280 ms   281 ms   270 ms  uswpho-2-t3-gw.customer.alter.net [157.130.232.14]

 14   310 ms   340 ms   311 ms  20.fa6-0.phnx-agw2.phnx.uswest.net [206.80.192.217]

 15   240 ms   261 ms   250 ms  100.fa0-0-0.phnx-6400-gw1.phnx.uswest.net
[209.181.135.96]

 16  100.fa0-0-0.phnx-6400-gw1.phnx.uswest.net [209.181.135.96]  reports: Destination
host unreachable.

Trace complete.
```

Hack'a'Tack probe: Hack'A'Tack is a Win9.x trojan program, but it may also work on WinNT, but this is undocumented. The trojan server portion is named "expl32.exe" and is sized at 236KB – released 5/16/99 2:49PM. It updates the WINDOWS directory. The default TCP ports used are 31785 and 31787. The default UDP ports are 31789 and 31791 are used to establish the covert channel connection between the "client" and "server". Once installed, it is rerun every time the computer is started by means of an entry under the "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" branch in the Registry. Hack 'a' Tack was written by Da SuckA and The Bart33. Further detail is available at: *http://www.commodon.com/threat/threat-hack.htm*. The *www.doshelp.com/trojanports.html* link identifies TCP port 31789 usage, contrary to UDP port 37189 as documented online at commodon. Also, "doshelp" indicates usage of UDP port 31790 without corroboration from "commodon," but modification and custom usage may be the rationale for this additional port. Further description of Hack'A'Tack is provided by *http://www.xploiter.com/security/hackattack.html*, and has some minor discrepancies in terms of the executable server name and size, but elaborates the scope of features somewhat better, as follows:

The delivered zip file called Hack'a'Tack contains:

- server.exe - server size is 235k – release date unknown;
- Hack'a'Tack.exe - client size is 293k – release date unknown; and
- Readme.txt size and release date unknown.

The Trojan source is written in Delphi and is Medium Risk. Features include:
- FTP services
- Transmit IP:
- IP-Scanner
- General Information, i.e. Current User, Country, Time, OS and CPU.
- Send Messages:
  - Open/Close the CDROM.
  - Hide/Show the taskbar.
  - Disable/enable the monitor.
  - Disable keys Swap and click mouse buttons.
  - Set/freeze the cursor at a position you can adjust by coordinates.
  - Window Events allowing you to kill, focus, hide, show and rename a process.
- Remote server clipboard viewing.
- Send text to the active server window upon demand or at periodic intervals.
- Remote Boot Operations, i.e. shutdown, reboot, power-off and logoff the remote computer.
- Get Passwords.
- Keyspy.
- Filemanager
  - Make Screenshot.

BID detected two NetSphere probes. The first consisted of a series of 55 TCP port probes all targeted against the NetSphere Trojan, attempting to activate it on the numerous ports indicated. The attack originated from outside the *Cox@Home* domain, but was backtraced using *tracert*. The findings of that and the second probe attempt are presented below. Both were unexpected, fast, and are considered hostile, especially considering that first one went after 55 ports is a systematic and quick probe attack, with little fanfare and noise. The second was attributed to a return visit from the HSE-Montreal Point-To-Point dialer seen earlier in the analysis. It was not retraced as the other trace was already inconclusive and unproductive:

- 193.159.78.212 with the BID backtrace name "pC19F4ED4.dip.t-dialin.net";
- 216.209.209.234 with the BID backtrace name "HSE-Montreal-ppp101429.simpatico.ca".

The 193.159.78.212 address was *tracert*'ed back to a German customer of Deutsche Telekom, at: http://www.dtag.de/dtag/ipl2/cda/t1/, a Bonn, FRG based Internet provider, whose services include the entire range of telecommunications services. As can be seen from the trace, the path taken back to the subnet domain of origin was completed and the source there verified. A copy of the trace is provided below:

```
Tracing route to pC19F4ED4.dip.t-dialin.net [193.159.78.212]

over a maximum of 30 hops:


  1   <10 ms    10 ms   <10 ms  r1-fe0-0-100bt.chnd1.az.home.net [24.1.208.1]

  2    20 ms   <10 ms    20 ms  10.0.208.1

  3   <10 ms    10 ms   <10 ms  c1-pos5-0.phnxaz1.home.net [24.7.72.5]

  4    10 ms    10 ms    20 ms  c1-pos7-0.sndgca1.home.net [24.7.65.134]

  5    21 ms    40 ms    20 ms  c1-pos1-0.anhmca1.home.net [24.7.64.69]

  6    20 ms    30 ms    20 ms  24.7.65.165
```

```
 7    30 ms    30 ms    40 ms  sl-gw8-sj-0-3.sprintlink.net [144.232.192.157]
 8    20 ms    20 ms    20 ms  sl-bb11-sj-7-0.sprintlink.net [144.232.3.113]
 9    20 ms    20 ms    30 ms  sl-bb10-sj-9-0.sprintlink.net [144.232.3.25]
10    70 ms    70 ms    80 ms  sl-bb12-rly-5-0.sprintlink.net [144.232.9.217]
11    70 ms    70 ms    70 ms  sl-fb1-rly-9-0.sprintlink.net [144.232.9.26]
12    80 ms    70 ms    90 ms  208.30.208.10
13    70 ms    70 ms    81 ms  NYC-gw12.USA.net.DTAG.DE [194.25.6.101]
14   201 ms   150 ms   160 ms  HH-gw13.HH.net.DTAG.DE [194.25.6.93]
15   150 ms   141 ms   150 ms  HH-gw13.HH.net.DTAG.DE [212.185.9.179]
16   151 ms   150 ms   140 ms  H-gw13.H.net.DTAG.DE [194.25.121.189]
17   160 ms   150 ms   180 ms  DO-gw13.DO.net.DTAG.DE [194.25.120.209]
18   150 ms   150 ms   160 ms  DO-gw12.DO.net.DTAG.DE [62.156.140.89]
19   150 ms   150 ms   150 ms  BO-gw12.BO.net.DTAG.DE [194.25.120.178]
20   150 ms   160 ms   151 ms  E-gw12.E.net.DTAG.DE [194.25.120.225]
21   140 ms   150 ms   151 ms  E-rg1.E.net.DTAG.DE [212.185.8.85]
22     *        *        *     Request timed out.
23     *        *        *     Request timed out.
24     *        *        *     Request timed out.
25     *        *        *     Request timed out.
26     *        *        *     Request timed out.
27     *        *        *     Request timed out.
28     *        *        *     Request timed out.
29     *        *        *     Request timed out.
30     *        *        *     Request timed out.

Trace complete.
```

Summary of Fingerprints: Fingerprinting is a technique to glean valuable information about a host by forming crafted packets that are used to query the TCP/IP stack implementation of different OSs. The definitive whitepaper on fingerprinting, with numerous examples, is maintained for state-of-the-art currency by fyodor, the author of nmap, at: *http://www.insecure.org/nmap/nmap-fingerprinting-article.txt*.

Nmap OS Fingerprint: "Black Hat" scanning is a technique whereby an unusual combination of TCP flag options are crafted by tools, such as nmap, to see how the system responds. Usually, the attacker is trying to identify the victim's operating system, and the accuracy of the version descriptions are often quite astonishing.. This information can then help the attacker determine which weaknesses exist on that system. Knowing the versions that have exploitable weaknesses is critical to attack success. The attack planning examines known vulnerabilities of specific OS versions to pinpoint specific exploits, in hope that vendor supplied patches have not been installed. As newer OS versions incorporate prior patches for older vulnerabilities, newer ones are always discovered and distributed among the hack community, so it's a

never-ending cycle of attack/defense and monitoring to fingerprint soft targets that have not fixed known and corrected problems on their systems.

BID classifies an nmap OS fingerprint from a TCP OS fingerprint by its association with nmap tool characteristics. The nmap parameters seen consist of the TCP destination port objective, and the following composite set of TCP flag options: S (SYN), F (FIN), R (RESET), P (PUSH), A (ACK), U (URGENT), 4 (low-order unused bit), 8 (high-order unused bit options. The TCP options from the offending frame are displayed as "option-value", separated by commas. No-ops are not displayed.

BID detected one nmap OS fingerprint attack, consisting of a series of 5 attacks against TCP port 22 (SSH) and port 43035, a unassigned IANA port, and one not currently known to be used as a default trojan port. It is noteworthy to say the nmap OS fingerprint was only one of the first four of these attacks, all of which were fired within the space of 5 seconds. The last was seen firing within within 10:47 of the start of the first in the series. Needless to say, these were all considered very hostile.

The first attack seen was a series of 8 TCP Fin scans against TCP ports: 22 (SSH Remote Login), 53 (DNS), 2049 (NFS), and 2222 (Rockwell CSP2). The second attack was a series of 4 TCP "ack" pings against TCP ports: 22 (SSH Remote Login) and 43035 (unassigned). The third attack was the nmap OS fingerprint whose *tracert* is shown below. The fourth attack was the TCP OS fingerprint described separately in the next section. The fifth attack was a series of 6 TCP port probes against TCP port 22 (SSH Remote Login), and this has also been previously described in the TCP port probe section.

The nmap OS fingerprint attack originated from outside the *Cox@Home* domain, and was backtraced using *tracert*. The findings of that attempt are presented below:

- 203.251.87.5 with no backtrace name from BID.

The 203.251.87.5 address was *tracert*'ed back to a S. Korean customer of the Korea Telecom (KORNET), at: http://www.kornet.net/serv/eng_m/new-i.html, a Seoul, KO based Internet provider, whose services include high-speed data communications and Internet services. As can be seen from the trace, the path taken back to the subnet domain of origin was completed and the source there verified. A copy of the trace is provided below:

```
Tracing route to 203.251.87.5 over a maximum of 30 hops


  1   <10 ms   <10 ms    10 ms  r1-fe0-0-100bt.chnd1.az.home.net [24.1.208.1]
  2   <10 ms    10 ms   <10 ms  10.0.208.1
  3   <10 ms   <10 ms    10 ms  c1-pos5-0.phnxaz1.home.net [24.7.72.5]
  4    10 ms    20 ms    10 ms  c1-pos7-0.sndgca1.home.net [24.7.65.134]
  5    10 ms    20 ms    10 ms  c1-pos1-0.anhmca1.home.net [24.7.64.69]
  6    20 ms    20 ms    20 ms  24.7.65.165
  7    20 ms    20 ms    30 ms  sl-gw8-sj-0-3.sprintlink.net [144.232.192.157]
  8    30 ms    20 ms    20 ms  sl-bb11-sj-7-0.sprintlink.net [144.232.3.113]
  9    30 ms    20 ms    30 ms  sl-bb10-stk-10-0.sprintlink.net [144.232.9.166]
 10    20 ms    20 ms    30 ms  sl-bb12-stk-9-0.sprintlink.net [144.232.4.114]
 11    30 ms    30 ms    20 ms  gip-stock-4-pos-oc3.gip.net [144.232.4.126]
 12   201 ms   200 ms   200 ms  204.59.163.142
```

```
 13   190 ms   201 ms   190 ms  apgate-gsr1-ge10.kornet.net [210.183.28.70]

 14   191 ms   200 ms   190 ms  center-gsr3-ge1.kornet.net [168.126.16.60]

 15   190 ms   200 ms   190 ms  taegu1-center3-622M-2.kornet.net [168.126.109.102]

 16   200 ms   211 ms   190 ms  203.251.82.246

 17   210 ms   201 ms   200 ms  210.105.111.14

 18   220 ms   211 ms   220 ms  203.251.87.5


Trace complete.
------------------------------------------------------------------------------
```

TCP OS Fingerprint: TCP OS fingerprints are separately classified by BID on the basis off their lack of association with nmap tool characteristics, or the Whisker CGI scanner supported by rain.forest.puppy.

BID detected two TCP OS fingerprint attacks, the first of which was seen to originate from the same S. Korean source as those identified in the concerted attack described in the nmap OS fingerprint section, above. The second attack was seen during the last week of monitoring and also originated from outside the *Cox@Home* domain. Both are considered to be very hostile.

The first TCP OS fingerprint attack originated from S. Korea was backtraced using *tracert*., the results of which are provided above, and have not been reproduced again for his reason:

- 203.251.87.5 with no backtrace name from BID.

The first TCP OS fingerprint attack was seen to be a series of 9 TCP port probes against TCP ports: 22 (SSH Remote Login) and 43035 (unassigned by IANA and not found to be a known trojan default port). There seems to be an undue level of attention being paid to port 43035 in several of the S. Korean attacks witnessed, and this is very disturbing as it may mean an attempt to probe other channels besides SSH for potential exploitation of trusted relationships between subnets or networks in the Far East. However, it is noteworthy that throughout the series of probes the TCP flag bits were alternately set to 0, 41, and 43. Using the urg|ack|PSH|RST|SYN|FIN order of TCP flags, the specific TCP OS fingerprint decoding reveals the following:

- with null or no flags set (0);
- with the urg/PSH/FIN (41=101001 bits) flags set; and
- with the urg/PSH/SYN/FIN (43=101011 bits) flags set.

But these TCP header flag settings are impossible because:

- at least one bit is required to identify the purpose of the packet;
- the PSH bits are set and the ack bit is not set;
- the PSH bit is inconsistent with either SYN or FIN bits set; and
- the SYN and FIN bit are simultaneously set.

Nmap uses the following impossible bit configurations (**bolded** below as well to relate to those actually seen, above) in TCP flags, in concert with the value of the TCP options and the open/closed state of the port being checked, to determine the OS:

Flags	TCP options	Target Port
S	yes	open
Null	**yes**	**open**
SFUP	**yes**	**open**
A	yes	open
S	yes	closed

A	yes	closed
FUP	**yes**	**closed**

Again, the hostile intent behind this fingerprinting is to identify the operating system type, using the specific nature of the response and the specific nature of each OS's response to each illegal and/or inconsistent TCP flag conditions shown. These flag values are deliberately crafted to elicit determined responses from the OS under attack.

The second TCP OS fingerprint attack originated from outside the *Cox@Home* domain, and was backtraced using *tracert*. The findings of that attempt are presented below:

- 63.70.25.53 with no backtrace name from BID.

Only one instance of the TCP OS fingerprint attack was seen and it was directed against TCP port: 111 (SunRPC), using an inconsistent/impossible TCP flag bit value of 3, decoding to the simultaneous setting of the SYN and FIN flags. As described in the first TCP OS fingerprint analysis above, this was deliberate and intended to get the target OS to reveal intrinsic data about itself, in hopes of furthering attack plan intelligence gathering for use at a later time.

The 63.70.25.53 address was *tracert*'ed back to a NY City customer of the UUNET, an MCI company, at: http://www.ALTER.NET/, a global based Internet provider, whose services include high-speed data communications and Internet services. As can be seen from the trace, the path taken back to the subnet domain of origin was completed and the source there verified. A copy of the trace is provided below:

```
Tracing route to 63.70.25.53 over a maximum of 30 hops


  1   <10 ms    10 ms   <10 ms  r1-fe0-0-100bt.chnd1.az.home.net [24.1.208.1]
  2   <10 ms   <10 ms   <10 ms  10.0.208.1
  3   <10 ms   <10 ms    10 ms  c1-pos5-2.phnxaz1.home.net [24.7.74.5]
  4    10 ms    10 ms    10 ms  c1-pos7-0.sndgca1.home.net [24.7.65.134]
  5    20 ms    10 ms    20 ms  c1-pos1-0.anhmca1.home.net [24.7.64.69]
  6   170 ms    20 ms    20 ms  24.7.65.165
  7   430 ms   421 ms   421 ms  148.ATM1-0.BR2.sjc1.ALTER.NET [137.39.91.25]
  8   431 ms   421 ms   430 ms  154.ATM3-0.XR1.SJC1.ALTER.NET [152.63.51.174]
  9   420 ms   401 ms   411 ms  193.at-2-0-0.TR3.SCL1.ALTER.NET [152.63.48.190]
 10   501 ms   481 ms   480 ms  207.ATM6-0.TR1.NYC1.ALTER.NET [152.63.3.149]
 11   470 ms   481 ms   471 ms  199.ATM7-0.XR1.NYC4.ALTER.NET [146.188.179.21]
 12   471 ms   481 ms   490 ms  189.ATM9-0-0.GW1.NYC6.ALTER.NET [146.188.178.161]
 13  1081 ms  1042 ms  1031 ms  63.71.58.1
 14  1082 ms  1112 ms  1101 ms  63.71.58.98
 15  1072 ms  1072 ms  1061 ms  63.71.58.1
 16  1041 ms  1052 ms  1051 ms  63.71.58.98
 17  1081 ms  1092 ms  1061 ms  63.71.58.1
 18  1041 ms  1102 ms  1021 ms  63.71.58.98
```

```
 19  1021 ms  1042 ms  1021 ms  63.71.58.1

 20  1041 ms  1052 ms  1121 ms  63.71.58.98

 21  1021 ms  1032 ms  1051 ms  63.71.58.1

 22  1101 ms  1162 ms  1052 ms  63.71.58.98

 23  1031 ms  1072 ms  1021 ms  63.71.58.1

 24  1001 ms   972 ms  1001 ms  63.71.58.98

 25  1031 ms  1042 ms  1031 ms  63.71.58.1

 26  1031 ms  1012 ms  1061 ms  63.71.58.98

 27  1001 ms   971 ms   962 ms  63.71.58.1

 28  1002 ms  1001 ms   972 ms  63.71.58.98

 29  1002 ms  1011 ms  1012 ms  63.71.58.1

 30  1012 ms  1051 ms  1021 ms  63.71.58.98


Trace complete.
```

Summary of Attacks: Many attacks seen during this 60 day observation period rely upon IP spoofing, the creation (crafting) of TCP/IP packets using an alibi IP address. This is done to confound attempts to gain the true identify of the originating source and track the attacker. Routers use "destination IP" addresses to forward packets throughout the Internet, ignoring "source IP" addresses. Only the destination address uses source IP addresses to send replies to the originator. However, it is commonly thought that IP spoofing can be used to hide your IP address while surfing the Internet, chatting on-line, and sending e-mail, but this is a misconception and as a general true, it is not true. Forging the source IP address alone without also hijacking the address of a known, live, and operating source IP addresses host causes the responses to be misdirected to some blind destination. This means that a normal network connection is never established in any sense. The fact that many network attacks do not require a normal network connection to be successful means that blind spoofing is all the capability that is necessary for the attack to work. Examples of spoofing attacks are:

- man-in-the-middle - packet sniffs on link between the two end points, pretending to be one end of the connection;
- routing redirect - redirects routing information from the original host to the hacker's host (this is another form of man-in-the-middle attack);
- source routing - redirects individual packets by hackers host;
- blind spoofing - predicts responses from a host, allowing commands to be sent, but can't get immediate feedback;
- flooding - SYN flood fills up the receive queue from random source addresses; and
- flooding - smurf/fraggle spoofs a victims address, causing everyone to respond to the victim.

All these enumerated attacks classified by BID use spoofing, but only some of them were captured and recorded during the 60 day period of this report, and they are detailed separately below with specifics:

- Land
- Teardrop
- NewYear
- SynDrop
- TearDrop2
- Bonk

- Boink
- Fragment Overlap
- Pind of Death
- IP Source Route
- Ping Storm
- Smurf
- ICMP unreachable advertisement
- UDP port loopback
- Snork
- Fraggle
- SYN flood
- DNS spoof

Fraggle Attack: This is not an attack against your system. Instead, it may be an attempt to "bounce" traffic off your system in order to overload someone else's Internet connection. BID triggers this alert from discovery programs attempting to map out the network, and for this reason they could reflect a network neighbor merely trying to find all their virtual cyberspace neighbors. The idea of finding one's neighbors on the network is called "mapping." Internet Protocol v4 supports a "broadcast" feature that allows someone to send a single "packet" to hundreds of computers on the same "subnet." The intended purpose of broadcasts is to make discovery easier. It allows your machine to announce itself to your neighbors, and it allows people scanning the network to easily find other machines. BID advice uses the SONAR analogy of a submarine to help understand this concept.

A cable-modem ISP subscriber may try to "map" their network segment by running programs to send out requests to elicit echo response, and build a list of everyone that replies. By varying the broadcast subnet mask used, accurate maps of a network can be systematically generated for each subnet. Machines that are "always on" and visible are very efficiently mapped in this manner using only one echo request packet. Several types of mapping programs have this capability, and they can be used for "good or evil" intentions. This is where the distinction between authorized and unauthorized use separates the "White Hats" from the "Black Hats."

This mapping process can be abused in a special attack known as "fraggle". The attacker isn't trying to attack you, but instead is using a feature known as "spoofing" to attack some third party. The way it works is that the attacker pretends to be the victim and sends out these echo packets to a subnet. Everyone (whose machine is on and so configured) will respond back to the victim. All the victim sees is that everyone from a subnet is flooding him/her with numerous echo responses. This attack was common during the Kosovo crisis. Pro-Serbian hackers would send out these echoes to many places on the Internet, spoofing U.S. and NATO sites. These sites where then overloaded with all the responses, taking the sites off line. When the BID intrusion-detection engine triggers this detection, it may be due to someone using the Cox network neighborhood as a staging point for these attacks. The firewall subsystem of BID blocks these incoming packets by default (when set to "Cautious" or higher). This installation is set at "Paranoid." If the attacker is using the Cox network for discovery purposes, then the attacker will not see this machine machine on that subnet. If they are using the Cox network to trigger floods against someone, then this machine will not participate in the flood.

Cable-modems and DSL lines are often placed in a common "broadcast domain" with thousands of other users. There is a good chance that one of these users will be running a discovery program in the background. BID provides a configuration tip disable these alert notices, but will have no effect on blocking of the packets.

Specific echo packets that can be sent at a machine in order to trigger echoes include:

- ICMP Echo - used in the standard 'ping' command;
- ICMP Echo - used in the smurf attack, similar to fraggle;
- UDP Echo – used to reflect traffic back to the sender.
- UDP Echo – used as the primary packet used in fraggle;

- chargen - returns random traffic back to the sender;
- daytime - returns the current time back to the sender;
- quotd - returns a "quote of the day" or "fortune cookie" back to the sender.

More information on "smurf" IP Denial-of-Service Attacks can be found in CERT: CA-98.01.smurf, at: *http://www.cert.org/advisories/CA-98.01.smurf.html*.

BID detected five possible Fraggle attacks using the broadcast netmask of 255.255.255.255. The first (2/18/00) consisted of a series of 3 broadcasts to port 7 (echo TCP/UDP) from source port 1026-1028, in sequence. The second (2/24/00) was seen to use 2 broadcasts to port 7 from source port 34816. The third (2/27/00) broadcasted twice, to port 7 from source ports 1025, and again to port 7 from source port 1030. The fourth (3/8/00) was from the same IP address as the first, only this time using one broadcast to port 7 from source port 1029. The fifth (3/16/00) was from the same IP address as the first, only this time using 2 broadcasts to port 7 from source port 1029, and then 1030. No further attacks of this kind were recorded. All activity appears to originate from within the *Cox@Home* domain, and all is suspicious and unexplained, especially that repeated broadcast pattern seen from 24.1.213.121. This one was traced and considered hostile, due to the persistent nature of the repeating echo packet behavior.

- 24.1.213.121 with the BID backtrace name “cx59418-a.chnd1.az.home.com”;
- 24.1.216.128 with the BID backtrace name “cx385650-a.chnd1.az.home.com”;
- 24.1.217.56 with the BID backtrace name “cx790440-c.chnd1.az.home.com”;
- 24.1.213.121with the BID backtrace name “cx59418-a.chnd1.az.home.com”;
- 24.1.213.121with the BID backtrace name “cx59418-a.chnd1.az.home.com”.

As can be seen from the trace, the path taken back to the subnet domain of origin was completed and the source there verified. A copy of the trace is provided below:

```
Tracing route to cx59418-a.chnd1.az.home.com [24.1.213.121]

over a maximum of 30 hops:


  1   <10 ms    10 ms   <10 ms  r1-fe0-0-100bt.chnd1.az.home.net [24.1.208.1]

  2    10 ms    10 ms   <10 ms  cx59418-a.chnd1.az.home.com [24.1.213.121]


Trace complete.
```

Smurf Attack: BID classifies the Possible Smurf-amplifier attempt from an ICMP echo frame sent to a subnet address (x.x.x.0 or x.x.x.255). This may cause a flurry of echo responses which can overwhelm the network or the systems involved. This smurf attack uses IP spoofing in order to broadcast pings to an "amplifier" in order to overwhelm the victim with responses. This is an attempt to use your network as a "smurf amplifier". For example, somebody on a cable-modem segment can send out a broadcast ping to his/her neighbors while spoofing the IP address of a victim. All network neighbors then respond to that victim instead of the IP spoofing address, overloading the victim's link. For the price of one packet from a smurf attack IP address, thousands of packets are sent to the victim in hopes of incapacitating the victim. False positives are sometimes triggered from cable-modem subscribers sending out broadcasts over their local network segment. This can also happen on corporate intranets when insiders do broadcast inside their own domain. While this doesn't indicate an attempt to use your network as an amplifier, it does indicate that somebody is attempting discovery operations from outside or inside your network.

BID detected four varieties of possible Smurf attacks using 5 broadcast netmasks, as described below: The first (x.255.255.255) variety were seen from the following source IP addresses:

- 24.6.173.205 with the BID backtrace name "cx256452-b.chnd1.az.home.com;"

The second (x.x.x.255) variety were seen from the following source IP addresses:

- 24.1.214.98 with the BID backtrace name "cx400222-c.chnd1.az.home.com";
- **24.1.238.153(2)** with the BID backtrace name "cx869790-a.chnd1.az.home.com";
- 24.1.209.33 with the BID backtrace name "cx84142-a.chnd1.az.home.com";
- 24.1.216.93 with no backtrace name from BID;
- 24.6.164.235 with no backtrace name from BID;
- **24.11.88.205(2)** with no backtrace name from BID;
- **24.6.166.6(2)** with the BID backtrace name "cx914659-a.chnd1.az.home.com";
- 192.6.166.6 with no backtrace name from BID;
- 24.14.30.105 with the BID backtrace name "cx192679-a.chnd1.az.home.com";
- **24.11.88.205** with the BID backtrace name "cx253779-a.chnd1.az.home.com;
- ;24.1.208.33 with the BID backtrace name "proxy1.chnd1.az.home.com";
- 24.1.214.36 with the BID backtrace name "cx287142-a.chnd1.az.home.com";
- 24.1.217.6 with the BID backtrace name "cx716795-a.chnd1.az.home.com";
- 24.8.68.75 with the BID backtrace name "cx206832-a.chnd1.az.home.com".

The third (x.x.x.0) variety were seen from the following source IP addresses:

- 24.1.213.78 with no backtrace name from BID;
- 24.1.210.76 with no backtrace name from BID;
- 24.1.217.8 with the BID backtrace name "cx325976-a.chnd1.az.home.com".

The fourth (0.0.0.0) variety were seen from the following source IP addresses:

- 169.254.249.179 with no backtrace name from BID;
- 169.254.200.61 with no backtrace name from BID;
- 169.254.131.41 with no backtrace name from BID;
- 24.1.209.57 with no backtrace name from BID;
- 169.254.66.133 with no backtrace name from BID;
- 169.254.204.223 with no backtrace name from BID;
- 24.15.55.116 with the BID backtrace name "cx191920-a.chnd1.az.home.com";
- 24.1.208.154 with the BID backtrace name "cx30095-a.chnd1.az.home.com";
- 169.254.169.95 with no backtrace name from BID;
- 24.11.87.230 with the BID backtrace name "cx578073-a.chnd1.az.home.com".

Much, but not all, of the activity appears to originate from within the Cox@Home domain, and all of it is suspicious and unexplained, especially that repeated by those source IPs that are seen initiating the smurf broadcast patterns seen. Three examples of this are identified by the bolded and underlined **IPs** with the number of times seen to repeat smurf'ing behavior indicated in parentheses. One particular instance of this was seen using a name which was not backtraced by BID the first two occasions it was observed engaging in questionable activity on the Cox network. The last time, however, a name was discovered and reported, so this last instance constitutes the third occurrence, but as it's name was learned, it was not identified in parentheses as were the first two instances where no name was found! For this reason, the worst offender was tracert'ed, and the results are provided below. As can be seen from the trace, the path taken back to the subnet domain of origin was completed and the source there verified. A copy of the trace is provided below:

```
Tracing route to cx253779-a.chnd1.az.home.com [24.11.88.205]

over a maximum of 30 hops:


  1    10 ms    10 ms    10 ms  r1-fe0-0-100bt.chnd1.az.home.net [24.1.208.1]
```

```
  2   100 ms    30 ms    11 ms  cx253779-a.chnd1.az.home.com [24.11.88.205]
```

Trace complete.

TCP "SYN" Flood: This attack attempts to create a Denial of Service (DoS) or availability denial based on an overloading of the number of connection requests that a host can manage. The SYN flood attack does this by sending TCP connections requests faster than a machine can process them. The attack typically proceeds as follows:

- attacker creates a random source address for each packet;
- SYN flag is set in each request packet to open a new connection to the server from the spoofed IP address (source IP addresses used by SYN floods are almost always spoofed);
- victim responds to spoofed IP address, then waits about 3 minutes for confirmation that never arrives;
- victim's connection table fills up waiting for replies;
- after table fills up, all new connections are ignored, resulting in a "slow-down" of network performance or a total loss of responsiveness;
- legitimate users are ignored as well, and cannot access the server;
- once attacker stops flooding server, it usually goes back to normal state (SYN floods rarely crash servers)

Newer versions of OSs manage resources better, making it more difficult to overflow tables, but are still are vulnerable to some degree. SYN flood can be used as part of other attacks, such as disabling one side of a connection in TCP hijacking, or by preventing authentication or logging between servers. The SYN flood attack sends TCP connections requests faster than the system can process them. There are cases where BID triggers this detection as a false-positive, such as whenever a large number of SYN packets are seen after a busy web-site has been unavailable for a few minutes, and once brought back online, releases the connection events that have been waiting for the system to become available. CERT-Advisory CA-96.21, *http://www.cert.org/advisories/CA-96.21.tcp_syn_flooding.html*, speaks to the association of TCP SYN Flooding and IP Spoofing Attacks.

TCP session hijacking is when a hacker takes over a TCP session between two machines. Since most authentication only occurs at the start of a TCP session, this allows the hacker to gain access to a machine. A popular method is using source-routed IP packets. This allows a hacker at point A on the network to participate in a conversation between B and C by encouraging the IP packets to pass through its machine. If source-routing is turned off, the hacker can use "blind" hijacking, whereby it guesses the responses of the two machines. Thus, the hacker can send a command, but can never see the response. However, a common command would be to set a password allowing access from somewhere else on the net. A hacker can also be "inline" between B and C using a sniffing program to watch the conversation. This is known as a "man-in-the-middle attack". A common component of such an attack is to execute a denial-of-service (DoS) attack against one end-point to stop it from responding. This attack can be either against the machine to force it to crash, or against the network connection to force heavy packet loss.

BID detected 19 TCP SYN flood attacks, of which 16 were seen coming from destination IP address 63.226.21.232, and the other 3 from 62.226.21.239. All of this activity was confined to 3/12/00 and 3/16/00 and is closely related to many if not all of the companion detections reported on these two days.

In fact, the generated activity from both 63.226.21.x IP addresses (bolded) on 3/12, 3/16, and 3/18 can be correlated chronologically by the UTC dates/times indicated:

63.226.21.239 TCP SYN floods (3/12/00 @ 08:59);
63.226.21.239 TCP SYN floods (3/12/00 @ 09:00);
62.226.21.239 TCP port probes (3/12/00 @ 09:07);

63.226.21.232 TCP port probes (3/12/00 @ 09:07);
63.226.21.232 Traceroutes (3/12/00 @14:29);
63.226.21.232 TCP SYN floods (3/12/00 @ 14:31);
63.226.21.232 TCP SYN floods (3/12/00 @17:49-17:50);

63.226.21.232 Echo Storms (3/14/00 @ 04:21);

63.226.21.239 Traceroutes (3/16/00 @ 02:13);

63.226.21.232 Traceroutes (3/16/00 @ 02:22-02:31);
63.226.21.232 Echo Storms (3/16/00 @ 02:31);
63.226.21.232 TCP SYN floods (3/16/00 @05:21);

63.226.21.239 TCP SYN flood (3/16/00 @ 08:39);

63.226.21.232 Telnet port probes (3/16/00 @08:51);

63.226.21.239 RPC port probes (3/16/00 @ 08:51);

63.226.21.232 TCP SYN floods (3/16/00 @ 21:37);
63.226.21.232 Traceroutes (3/18/00 @01:13).

63.226.21.239 Traceroutes (3/18/00 @ 01:02).

More importantly, an ever present dialup connection from the **63.226.134.240** named **"qdslppp240.mpls.uswest.net"** is right in the middle of all this action! This is the one common thread to both attacks delivered through 63.226.21.232 and 63.226.21.239, on both 3/12 and 3/16! My analysis indicates this dialup USWest customer co-opted two user IPs on a different (but adjacent) USWest subnet to "bounce his attack" onto the 6 Cox subscribers indicated below:

24.1.216.237
24.1.238.151
24.1.11.88.170
24.9.115.57
24.8.69.68, and
24.11.88.2

The SYN flooding began the Day One attack on Sunday, March 12, 2000 at 1:59 A.M. MST. The first part of the attack was started through both USWest "reflector" (63.226.21.232 and 63.226.21.239) identities previously found to be useful for this purpose. They would shield the true identity of the culprit (operating on the adjacent subnet of USWest with a PPP dialup connection). After the SYN floods were given time to take affect, and cause a debilitating volume of bogus traffic to accumulate on the 6 Cox cable network victims, the simultaneous TCP port probing began in earnest. This was the second phase or active part of the attack and its intent was to use the Cox subscribers account identities to try to find vulnerable and available services around the network. Then the third phase of the attack would begin where the culprit used Telnet probing to attempt to discover what he though he could access remotely on Cox, using this masked Telenet conduit to launch host specific theft attacks, for example. This phase of the attack continued for up to 5 hours, allowing the attacker time to sift through the list of available ports and telnet to them, then to locate and remove any "interesting" data back out through use of the same telnet services. In this way the telnet could be used not only to search for any find the targeted data, but to move it over the covert connection pathway provided by the "reflectors" to the culprit's PPP switched connection. This would add a level of indirection to any attempts to trace the attack to its true origin throughout the break-in in progress.

Allowing time for the scripted attack to complete, ad then to complete any interactive data search and theft, the culprit completes his work in the allotted time and then runs a Traceroute barrage through the first "reflector" (63.226.21.232) to see if the Cox machines were settling back down after the initial SYN flood.

When satisfied after about two minutes of review that things looked too normal again, and that someone in authority might be looking into the cause of the Denial of Service (DoS) on the 6 Cox machines, the culprit decides to end the attack. To prevent someone from seeing a fresh "trail", the attacker unleashes a second SYN flood to confound anyone attempting to see through the haze to investigate the cause of all the trouble over the last 5.5 hours. With this, the Day One attack is over.

But the culprit isn't through just yet. On Monday, March 13, 2000 at 9:30 P.M. MST, an Echo Storm is hurled at the same 6 Cox machines to "stir the pot" in anticipation of the upcoming Day Two attack on Thursday.

Day Two begins on Wednesday, March 15, 2000 at 7:13 P.M. MST, with Traceroutes getting run through both "reflectors" to assure the culprit that the circuits through both "reflectors" to the 6 Cox hosts are reachable and usable. After spending no more than 20 minutes making sure the networks are ready, the culprit launches the Echo Storms through the first "reflector" at 7:31 P.M. MST. This renewed attack the Cox boxes makes the Cox Network Operations folks think they're seeing last Monday night all over again. Only this time, after 2:50 of Echo Storms, the SYN floods are unleashed through the first "reflector". The DoS against the same 6 Cox subscribers is raging once more. Just to keep everyone at Cox off-balance, in case they've figured out what went down Sunday morning, the culprit kicks in another round of SYN floods 3:18 after the first one, using the second "reflector" this time.

Allowing a mere 12 minutes for chaos to reign once more, the culprit simultaneously launches scripted Telnet probes through the first "reflector," and scripted RPC probes through the second. The time is now 1:51 A.M. Thursday, March 16, 2000. The attack has begun in earnest once more, and the culprit(s) use the next 12:46 to search through the telnet vulnerabilities found, and to sift through the SunRPC findings. Working to complete their telnet and RPC break-in to those exposed and easy targets, information is identified and bagged to delivery back to the attacker(s). Information found but not raided on Sunday is fair game too, as the culprit is free to revisit systems already visited one this week! By 2:37 MST, the "GAME IS OVER." The attacker finishes off the Cox boxes (and the rest of the network too) with "covering fire" from yet another SYN flood. After another 3:36, it's now 6:13 MST in Phoenix, and the "Black Hat" takes another Traceroute walk to see how things are doing once more after the floods are abating. After all, there may have been some other juicy tidbits that would be nice to pick up next week.

The fact that the attack came from a dialup PPP on USWest on Sunday night, and then again from the same source IP and name again on the following Monday and Wednesday, would be unexpected if the culprit were using a USWest DHCP connection. That this address is unvarying is a surprise, and for this reason the address was *tracert*'ed. As can be seen from the trace, the path taken back to the subnet domain of origin was completed and the source (**culprit**) there verified. A copy of the trace is provided below:

```
Tracing route to qdslppp240.mpls.uswest.net [63.226.134.240]

over a maximum of 30 hops:


  1    10 ms   <10 ms    10 ms  r1-fe0-0-100bt.chnd1.az.home.net [24.1.208.1]

  2   <10 ms    10 ms    10 ms  10.0.208.1

  3   <10 ms    10 ms    20 ms  c1-pos5-2.phnxaz1.home.net [24.7.74.5]

  4    30 ms    10 ms    10 ms  c1-pos7-0.sndgca1.home.net [24.7.65.134]

  5    20 ms    30 ms    20 ms  c1-pos1-0.anhmca1.home.net [24.7.64.69]

  6    30 ms    20 ms    20 ms  24.7.65.165

  7   271 ms   290 ms     *     148.ATM1-0.BR2.sjc1.ALTER.NET [137.39.91.25]

  8   310 ms   301 ms   310 ms  154.ATM3-0.XR2.SJC1.ALTER.NET [152.63.51.178]
```

```
  9   311 ms   340 ms   321 ms  192.at-2-1-0.TR2.SAC1.ALTER.NET [152.63.51.54]

 10   360 ms   351 ms   350 ms  127.at-6-1-0.TR2.CHI2.ALTER.NET [152.63.1.193]

 11   370 ms   361 ms   360 ms  296.ATM7-0.XR2.CHI4.ALTER.NET [152.63.65.77]

 12   380 ms   361 ms   340 ms  190.ATM8-0-0.GW2.MSP1.ALTER.NET [146.188.209.125]

 13   350 ms   321 ms   310 ms  uswmin2-gw.customer.ALTER.NET [157.130.106.210]

 14   300 ms   311 ms   310 ms  100.gig0-0-0.mpls-agw1.mpls.uswest.net [207.225.159.216]

 15   310 ms   361 ms   330 ms  100.fa0-0-0.mpls-6400-gw4.mpls.uswest.net
[207.225.140.99]

 16   400 ms   401 ms   390 ms  qdslppp240.mpls.uswest.net [63.226.134.240]


Trace complete.
```

Echo Storm: This is also referred to as a "ping' attack, due to the abnormally large number of ping packets which are seen in a short period of time. Pings are how one can tell if another machine on the Internet is running, and how long it takes to respond. This is a frequent attack against home users, and it occurs as they play games or chat. Under these conditions the attacker floods their machine with pings and slows their connection performance across the Internet, and debilitating their play or chat session as well. BID has found that a number of applications generate small storms on purpose in order to more accurately calculate the speed of that subscriber's connection. The instances of this seen by BID so far include VisualRoute, NetMeeting, and Net.Medic.

BID reported 12 Echo Storms during the period of the Day One and Day Two attacks described in the preceding section. The analysis is reflected in that discussion as these "storms" were felt to be linked to the same TCP SYN flood used to start the two attacks on Cox, and no other incidence of "storms" was seen during the 60 day monitoring.

Traceroute: Traceroute/tracert or tracert.exe (Unix/Win) is a network trace that permits somebody on the Internet to map the route taken by packets from their machine to your machine. It is utility program, and not an attack per se, but was placed into this category due to the apparently hostil intent of the spate of traceroutes seen in connection with the Echo Storm attack, described above. Other traceroutes that do not appear to have been a part of this attack scenario, are so indicated by the non-hostile or benign colo code used on the Summary spreadsheet.

The traceroute utility is widely used on the Internet in order to find the route between two machines. Imagine calling somebody on a phone and being able to look on a map to see a drawing of the exact route your phone call takes. The traceroute program performs this task, except it shows a "virtual" route through the Internet. A traceroute isn't very dangerous, as there is no way to break into your machine using this feature. However, it does help a hacker map out your Internet connection, and this is how it was used with Echo Storm, above. This information could possibly be used to compromise some other part of your connection. In the past this type of information was used by hackers in order to kick their victims off the Internet by forcing the nearest router to hang up the phone line, thereby breaking the switched modem connection.

Some BID users receive reports of their own machines sending out traceroutes which are flagged because BID thinks they are not normal traffic. This is especially true for application products, such as VisualRoute and Net.Medic, that do traceroutes as part of their normal activity, and may cause BID to interpret them as detections of a possible attack. For the purpose of this report, none of this should or did occur, as these applications are not installed.

The traceroute program pings each of the routers in the sequential order of encounter along the path to your machine. So if the gateway router to your system is outside your firewall, the trace will still lead right to your doorstep, disclosing your router's IP address and host name. Traceroute is an elegant program that works by sending out packets with short lifetimes in order to map the route to another machine. Each packet is given a slightly different Time-To-Live (TTL) value, so that when a router expires the packet, it sends back a notification. This process is done iteratively for each point along the path until the trace is ended, or until no timeout "lifetime" value large enough permits the entire trace to be completed. In this last case, the trace is never completed until the TTL lifetime is adjusted upwards and the trace restarted.

BID reported 33 Traceroutes altogether during the 60 day monitoring, and 23 of these alone were detected during the period of the Day One and Day Two attacks described in the preceding section. The analysis reflected in that discussion includes that necessary for these "traceroutes", as well as the "storms." It is the opinion of this analyst that the Traceroutes and Echo Storms are linked to the same TCP SYN flood used to stage the two SYN flood attacks described in the previous section. The remaining 10 instances of Traceroutes detected were felt to be benign, due to the fact that no other related hostile activity seemed to be occurring in parallel, thus freeing them from any suspicion.

Summary of Anomalies/Artifacts: "Black Hat" folks have been known to impersonate legitimate IP addresses on networks. In some cases these lead to detectable and reportable anomalies of an impending attack, or are in fact artifacts of attacks previously mounted and executed. In either case, they often reflect spoofed IP addresses of legitimate users and handled separately from the other categories listed above.

Echo Reply – Unsolicited: This condition is anomalous because the corresponding ICMP echo request message that is expected to result in an ICMP echo response/reply is never seen. This is not only an unexpected condition, but it can be dangerous. The danger lies in its deliberate use as a technique to scan systems behind a firewall. Incoming ICMP echo replies are usually allowed to pass through a firewall without being questioned, as most do not incorporate the dynamic state-based filtering capability needed to associate the reply with the previously issued request. Not doing so would cause every legitimate "ping" to fail. The problem is compounded whenever a packet arrives for a host that does not exist. For this condition the enterprise router may be configured to reply with an "unreachable" response message. If this is the case, then an attacker can systematically find all "unreachable" hosts that *do not* exist, and by inference, deduce the inverse mapping of all hosts that *do* exist. The potential for even greater harm arises from the possibility that echo replies can be used as a "covert channel" to communicate with "trojans" covertly implanted behind firewalls. This technique is similar to that seen used by various distributed denial of service (DDoS) tools to communicate remotely with trojan horse programs inside a protected network to launch coordinated attacks. CERT Incident Note IN-99-07, *http://www.cert.org/incident_notes/IN-99-07.html*, speaks to the threat of unsolicited echo replies as a tool associated with Distributed Denial of Service (DDoS) attacks.

BID detected 3 instances of an unsolicited echo replies, reporting each as an "Echo reply without request." All were source IP addressed as coming from 198.168.1.1, and were felt to be hostile as they were unexplainable and unexpected. The source IP in this case is normally associated with that one often seen used to configure routers on private enterprise networks (RFC1918). The recurring nature of the address throughout all private network domains makes it inconclusive if attempts are made to associate it with the source of an limited DoS Internet attack if used alone, or a smaller part of a larger attack plan if used in concert with other more sophisticated scenarios. This fact makes this particular address attractive to an attacker, because it introduces arguable doubt about the true identity of the source if an attack is traced to this location. In this case, it would appear to come from a legitimate router of a private enterprise outside of the *Cox@Home* network. It reality it is from outside the Cox domain, and except for the fact that BID was unable to backtrace any name for this address, which would have been expected had it really been a router, analysis says it's really not what it's pretending to be either! My analytical conclusion is that each of these separate attacks are hostile, rather stealthy due to the fact that they require a state-based detection capability (as the fact no requested echo reply exists that would account for their mysterious and silent appearance), and of rather high severity.

The first instance of this attack was noted on 4/3/00 and consisted of a series of 38 reply packets, which contained an id=0100. The second instance occurred the very next day, 4/4/00, and consisted of a series of 38 reply packets, which contained an id=0300. The last one seen occurred on 4/5/00 and consisted of 38 reply packets and an id=0200. Nothing further was seen which helped to understand to identity of the attacker.

Duplicate IPs: This condition means that two computers have been seen on your local segment using the same IP address. Every system must have its own IP address. A common attack is to take over the IP address of some other network entity, impersonate them on the network, and take over the management of their connections. In this way it may be possible for an attacker to take over the role of the local router. If this occurs, then everyone on the local network will be sending traffic through the "Black Hat" machine rather than their default router, and no one will be the wiser. The attacker will then be able to eavesdrop on everyone else's activities to steal passwords, credit card information, and anything else that flows over that channel. This technique is another example of covert channel establishment and use.

However, in version 1.9.25 of BID, the intrusion-detection engine sees lots of duplicate addresses on cable-modem segments, and it is more likely reason is because other Cox subscribers have simply mis-configured their machines with the wrong IP address, resulting in a "Duplicate IP" false alarm from BID.

BID disclosed 9 duplicate IP instances which were not felt to be hostile in any sense, and are attributed to configuration problems, as detailed below:

- 24.1.239.71(2) with the BID backtrace name "cx63175-d.chnd1.az.home.com";
- 24.6.166.1(2) with the BID backtrace name "r1-fe0-0.chnd1.az.home.com";
- 24.6.166.1(3) with no backtrace name from BID;
- 24.9.114.158 with the BID backtrace name "cx589913-b.chnd1.az.home.com";
- 24.6.173.110 with the BID backtrace name "cx5896490-a.chnd1.az.home.com".

Detects Witnessed by Type	Hostile Intent	Week Ending 22-Feb	28-Feb	6-Mar	13-Mar	20-Mar	27-Mar	3-Apr	10-Apr
Multicast/Discovery Scans:									
ICMP subnet mask request	Y	0	0	0	2	0	0	0	0
SNMP discovery from w/in Cox	Y	19	19	14	15	15	9	12	19
SNMP discovery from o/s Cox	Y	7	15	8	8	10	5	9	18
N/W Mgmt Scans, Probes, Pings:									
Whats Up scan	N	1	0	0	0	0	0	0	1
SMTP port probe	N,Y	0	1,0	0	0	0	0	0,1	0
NNTP port probe	N	4	2	1	2	4	0	3	4
PCAnywhere ping	N	1	2	0	0	1	2	2	1
Port Scans:									
TCP port scan	Y	2	0	0	0	7	1	0	0
TCP "FIN" scan	Y	0	0	0	0	1	0	0	0
Pings:									
Ping Sweep	Y	1	0	0	2	0	0	0	0
TCP "ack" ping	N,Y	0	1,0	0	0	0,1	0	0	0
Back Orifice ping	Y	3	0	2	2	0	1	2	0
Port Probes:									
PCAnywhere port probe	Y	0	0	0	0	1	0	0	0
FTP port probe	Y	1	1	3	3	1	2	1	2
UDP port probe	Y	0	1	2	0	0	0	1	46
TCP port probe	Y	16	15	10	22	12	8	7	4
Proxy port probe	Y	2	0	0	5	2	0	2	7
DNS port probe	N,Y	1,0	0	0	1,0	0,1	0,2	1,4	2,24
MSRPC port probe	Y	0	0	0	0,1	0	0	0,1	0
RPC port probe	N,Y	1,0	0,1	0	2,0	7,2	0	2,0	2,3
Telnet port probe	N,Y	0,3	0,2	0,1	0,1	6,5	0,1	0	2,0
Trojan Probes:									
NetBus port probe	Y	0	1	1	0	1	0	0	1
SubSeven	Y	3	2	3	0	6	1	1	5
TCP trojan (NetSphere) probe	Y	0	0	0	0	0	1	0	1
UDP trojan (Hack'a'Tack) probe	Y	0	0	0	0	0	0	0	1
Fingerprints:									

nmap OS fingerprinting	Y	0	0	0	0	1	0	0	0
TCP OS fingerprinting	Y	0	0	0	0	1	0	0	1
Attacks:									
Possible Fraggle attack initiated	N,Y	0,1	1,1	0	1,1	0	0	0	0
Possible Smurf attack initiated	Y	3	9	3	3	10	1	1	1
TCP "SYN" flood	Y	0	0	0	10	9	0	0	0
Echo storm	Y	0	0	0	0	12	0	0	0
Trace route	N,Y	2,0	1,0	0	7,0	0,23	0	0	0
Anomalies/Artifacts:									
Echo reply - unsolicited	Y	0	0	0	0	0	0	1	2
Duplicate IP address	N	1	0	1	0	6	0	0	1

Detects Witnessed by Type	Notes, Comments, or Other Significant Observations	Ports Used:
Multicast/Discovery Scans:		
ICMP subnet mask request	see more verbose details in report text	none used
SNMP discovery from w/in Cox	see more verbose details in report text	161
SNMP discovery from o/s Cox	see more verbose details in report text	161
N/W Mgmt Scans, Probes, Pings:		
Whats Up scan	see more verbose details in report text	none used
SMTP port probe	1st detect from ops-scan.home.net; 2nd is not!	25
NNTP port probe	all detects from ops-scan.home.net	119
PCAnywhere ping	all pings seen originated from w/in Cox domain	22, 5632
Port Scans:		
TCP port scan	see more verbose details in report text	numerous
TCP "FIN" scan	external scan against ports 42, 53, 2049, 2222	as noted
Pings:		
Ping Sweep	seen from inside/outside Cox domain	8
TCP "ack" ping	1st detect from w/in Cox on port 80; 2nd is not!	22, 43035
Back Orifice ping	2 variants seen; all but 1 used vport 31337	31666
Port Probes:		
PCAnywhere port probe	unauth attempt to gain command control blocked	5631
FTP port probe	seen from inside/outside Cox domain	21
UDP port probe	see more verbose details in report text	quite a few
TCP port probe	see more verbose details in report text	numerous
Proxy port probe	see more verbose details in report text	3128, 8080
DNS port probe	see more verbose details in report text	53
MSRPC port probe	all originated from 0.0.0.0 with no DNS name	135
RPC port probe	see more verbose details in report text	111
Telnet port probe	see more verbose details in report text	23
Trojan Probes:		
NetBus port probe	all tried activating name Netbus w/default port	12345
SubSeven	older Sub7s seen, but 2 instances of Sub7 2.1	12438
TCP trojan (NetSphere) probe	one IP probed 55 TCP ports for NetSphere!	numerous
UDP trojan (Hack'a'Tack) probe	one IP probed 1 TCP port for Hack'a'Tack!	31789
Fingerprints:		
nmap OS fingerprinting	1 detect used port 43035 and also port ->	22
TCP OS fingerprinting	1st detect used 22,43035; 2nd detect used port-> 111	

Attacks:		
Possible Fraggle attack initiated	3 detects from same IP inside Cox domain	7
Possible Smurf attack initiated	4 varieties of netmasks used	none used
TCP "SYN" flood	verbose detail of Day One and Day Two attacks	quite a few
Echo storm	verbose detail of Day One and Day Two attacks	none used
Trace route	verbose detail of Day One and Day Two attacks	none used
Anomalies/Artifacts:		
Echo reply - unsolicited	all were source IP addressed from 198.168.1.1	7
Duplicate IP address	affected IP addresses detailed in report text	none used