Kenneth Gallo February 20, 2001
Incident Report for GCIH

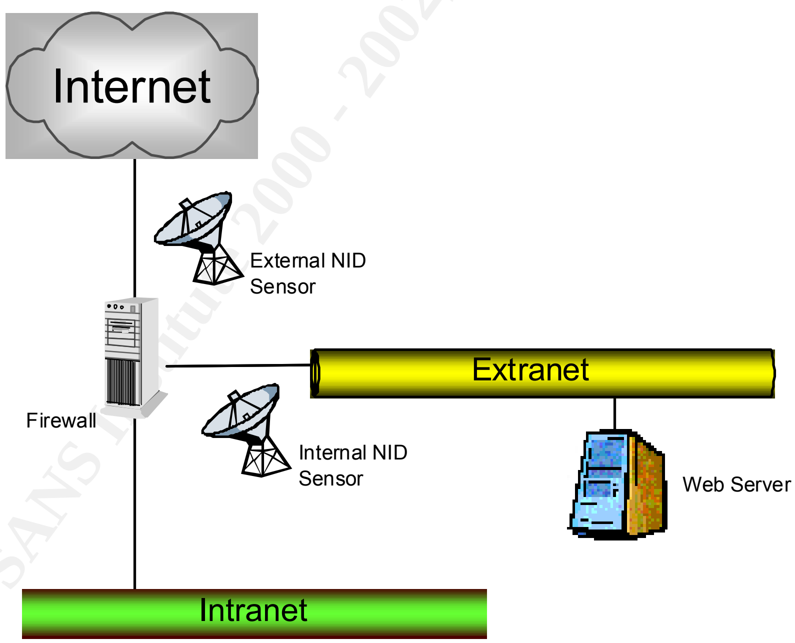
Option 1 – Illustrate an Incident

Executive Summary

It had been only two weeks since I installed the network intrusion detection system at this Fortune 500 firm.

October 27th, 2000 started out as a normal Friday. I decided to continue tuning the new network intrusion detection (NID) system. The NID had been producing the expected barrage of alarms that come with any such system, and I had to start sorting out the real events from the false ones. Before this system was installed, the company had been blissfully ignorant of the several attacks that took place against it daily.

I browsed through the past days' event logs, noting that the attempted port scans and other reconnaissance efforts against the company had continued unabated through the night. The external NID sensor saw many events, but few got through to where the internal NID sensor was placed. The firewall seemed to be doing its job.



Then I noticed something unusual, at 1:29 in the morning the internal NID sensor saw malicious packets leave one of our web servers. A second later the external NID sensor caught the same packets shooting outbound to an Internet address. According to the NID system, those packets came from a distributed denial-of-service tool called 'Mstream'.

We were apparently launching an attack against another company.

Analysis of an Incident

At first I could not believe what I was looking at. The NID system, the very same security system the company had installed to stop attacks from the outside, had instead detected an attack coming from the inside. Then it hit me: if our servers were attacking someone else, then the real attackers must have already compromised our servers.

Ironically, all of this was happening about six weeks before I was scheduled to attend a SANS conference course about incident handling. Luckily, I already had part of the course material, the "SANS Incident Handling: Step by Step" guide (1). This booklet became a critical tool in the days ahead.

Six Phases of Incident Handling

It is useful to give some background information here about incident handling. The "SANS Incident Handling: Step by Step" guide describes the process as a never ending exercise made up of six phases:

Phase 1) PREPARATION – This is the most difficult and time-consuming phase in the process. Preparation includes: defining a good security policy; educating users on that policy; hardening systems and networks; implementing security tools; documenting the key components and applications in the environment; creating an incident handling plan and team; gathering the supplies the team will need; getting management support, etc.

Phase 2) IDENTIFICATION – A security tool, or a user who notices that something is wrong, are what detect most potential incidents. When those events are reported the incident handling team must decide whether it is a true incident. If it is, then the team needs to identify the nature of the incident, its scope, and its severity.

Phase 3) CONTAINMENT – Once the incident has been identified, it (and the damage it may cause) must be contained. The incident handling team is deployed onsite to secure the area and the affected systems. The systems are carefully backed up and the passwords are changed. The team, together with systems owners and management, must determine the risk of continuing system operations.

Phase 4) ERADICATION – The incident handling team must determine the cause of the event, and what mechanism was utilized to bring it about. That cause must be removed, and the systems' defenses are improved to prevent similar incidents. A vulnerability analysis is conducted on the systems, and the most recent clean backup (if available) is located.

Phase 5) RECOVERY – The incident handling team and management must decide when to restore operations. When the systems are recovered, they are restored from a clean backup if one is available. If the backups cannot be trusted, the systems must be rebuilt.

Phase 6) FOLLOW UP – The final step is for the incident handling team to document the lessons learned so that a better job is done in the future. This work needs to be done very soon after the incident, and the recommended improvements need to be incorporated into the environment and the company's incident handling plan.

How the Incident was Handled

The scope of the incident had to be assessed quickly and its damage contained, the operation of a multi-billion dollar e-commerce site was at risk.

The SANS course on incident handling, which I attended several weeks after these events, taught me many important things about the proper methodology and techniques that should be followed. Some things were done correctly, and others could have been done better.

Phase 1 – Preparation
Long before the day of the incident, the company had been preparing itself by becoming more and more vigilant about security. There were numerous security initiatives taking place:

- Staffing – Before I joined the company there were only two people in the company's information security department, and neither was focused on the technical requirements of security. I was brought in to help the company identify and remedy the technical deficiencies.

- Vulnerability Analysis – Every month I performed a vulnerability analysis on every critical server and network device by scanning them using ISS Internet Scanner. The tool provides detailed reports that, while not always accurate, supplies key information about deficiencies in the company's security posture. Those reports were entered into a simple database to keep track of detected weaknesses and provide a trending analysis to objectively measure whether the company was improving.

- Server Hardening – The volume of reports generated by the vulnerability analysis led to a project to reduce those weaknesses by tightening down the servers. The project had just begun, but was already applying critical patches and disabling some dangerous services in the environment.

- Network Intrusion Detection - Realizing that server hardening alone would never be enough; the company had recently invested in an ISS RealSecure network intrusion detection (NID) system. The NID sensors sniff the network and analyze packets for predefined suspicious activity. We only had licenses for two sensors, so I installed one on the company's Internet connection, and the other on the extranet.

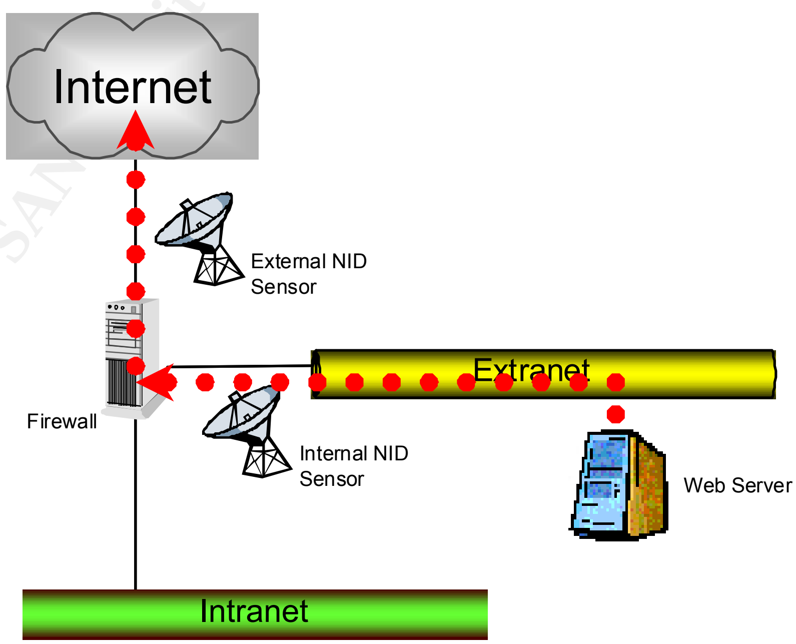
- Host Intrusion Detection – I had also recently started a project to look at host intrusion detection (HID) systems for the company. HID sensors reside on the protected hosts, analyzing audit logs and other activities for signatures of suspicious activity. One of the tools I was testing was Symantec's (formerly Axent) Intruder Alert software.

- Security Policy Enforcement – All the reactive defenses in the world don't do any good without some good proactive defenses. I had also just started a project to look at tools to semi-automate system audits to help enforce security policy. One of the tools I was testing was Symantec's (formerly Axent) Enterprise Security Manager.

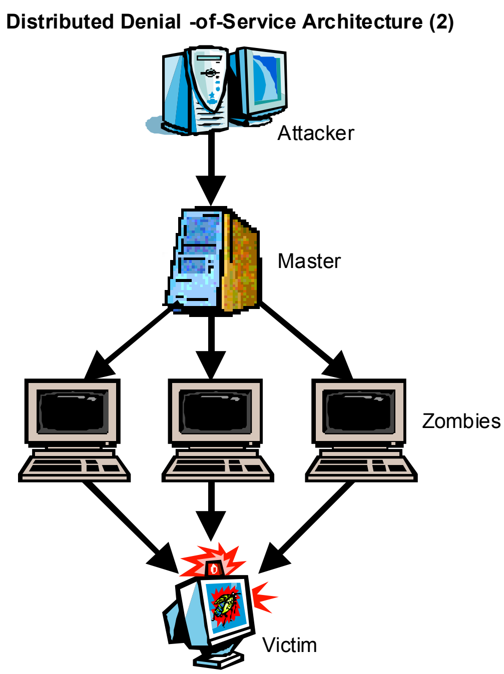
Despite all of the above, there was no incident handling plan. However, the company knew it was needed, and was sending me to SANS in the future so that I could develop it.

<u>Phase 2 – Identification</u>

On Friday October 27th at 1:30 in the morning one of the company's primary web servers sent what seemed innocuous packets to a company in Asia. As the packets left the extranet, a NID sensor placed there immediately detected the signature for Mstream Distributed Denial-of-Service packets and raised an alarm. A second later, as the packets where leaving the company, a second NID sensor on the Internet connection also saw the packets and sounded an alarm.



Unfortunately, the NID system was new, and wasn't yet hooked up to the company's paging or email system. Thus, the alarms were sounded but went unheard. The company would have to wait until morning before anyone knew what was going on.



At about 9:30 that morning I entered the data center to check on the NID system. After a few minutes of browsing through the event logs I noticed the Mstream events from the night before. I had never seen this event before, and certainly not any originating from our own networks.

The ISS RealSecure documentation described the signatures as belonging to Mstream Master, software that controls Mstream zombies on other hosts to commit distributed denial-of-service attacks. The documentation went on to say that installing an Mstream Master on a server requires root, and that there were no known false positives for the NID signature; in other words, it was not a false alarm (3).

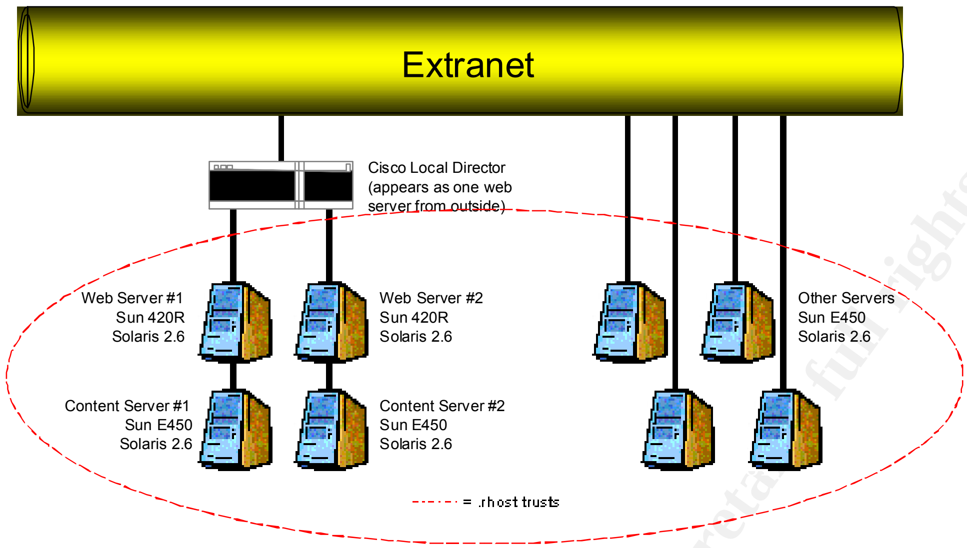
This meant one of two things: either one of the company system administrators had gone bad; or, more likely, someone had compromised one or more of our servers.

Phase 3 – Containment

Now that I had identified the symptoms, the threat had to be contained. I immediately informed my manager of the situation, and was requested to keep him apprised. I then grabbed Bob, a senior systems administrator at the company, and my copy of the "SANS Incident Handling: Step by Step" guide.

Bob and I went back into the data center, where I brought him up to speed on the NID system, what the sensors saw, and what it all meant.

Bob understood the seriousness of the situation. He then explained to me what systems and architecture we were dealing with: the web server was actually two separate Solaris 2.6 servers behind a Cisco Local Director load balancer; the two servers, in turn, mounted their content off of two other Solaris 2.6 servers; those content servers, in turn, had .rhost trust relationships with four other servers (represented by circle below).



Because Cisco Local Directors do not log which server they are pointing to at any given moment, there was no way to determine which server sent the Mstream packets. The web servers, the content servers, and the other servers added up to eight potential compromised boxes. We would need to check each and every one, and it would be a long night.

Bob and I took this opportunity to read the SANS guide. We found many good ideas, and decided to try following it for the rest of the incident. Management also liked the plan, and told us to 'do whatever is needed' to fix the problem.

As the company did not have any official incident handling program in place, there was no incident handling team or jump bag. Bob and I became that team but we did not have any special equipment. We mostly used a simple VT100 telnet emulator to command the affected systems.

Another piece of equipment we had was a brand new Sun 420R server, identical hardware to the existing web servers. It was scheduled to go into production later that week, but Bob and I had commandeered it. We kept it ready in case we would need to remove one of the existing web servers for forensic analysis.

The SANS guide recommended that we backup the affected systems. The Legato backup system used by the company could not back them up all at once, so Bob and I would have to bring each one down separately and have the operations personnel perform a full cold backup. The processes that were used are documented in Attachment A.

Bob and I began analysis on each system as it came back on after backup. We knew each server already had numerous vulnerabilities, many of them documented in my monthly vulnerability analysis. Finding which vulnerability may have been exploited seemed infeasible in the short amount of time we had. Additionally, a meeting with management had determined that the business placed a higher priority on getting service back up rather than forensic analysis or preservation for this incident.

Therefore the next best step for Bob and I was to change the root passwords on all the affected and nearby systems. As all the systems were Solaris, we did this by logging in as the root user and typing the standard "passwd" command.

After that we installed Symantec Enterprise Security Manager (ESM) on the servers. I was already looking at ESM as part of an evaluation project. The software operates through small agents placed on protected servers. Those ESM agents perform fairly thorough internal audits that include: checking for trojans; verifying patch levels; testing password strength; examining file ACLs; and hunting for malicious code, like Mstream.

Each ESM agent was ordered to perform an immediate and full audit (using the prepackaged Phase 3:c Strict ESM policy) of the systems after installation. Each audit took between 1 and 3 hours to complete. The reports were then consolidated on the ESM console.

The ESM audits detailed many of the vulnerabilities we expected, and a few we did not expect. However, ESM did not find any traces of Mstream or any other malicious code on the servers. This was a surprise, since we felt confident that Mstream packets had been seen originating from at least one of those servers.

<u>Phase 4 – Eradication</u>
As we were unable to find the Mstream binaries, we will never know for sure exactly what caused those malicious packets that morning. Left with symptoms created by an unknown cause, the next best thing we could do was prevent problems from happening again.

First, we thoroughly analyzed the ESM audit reports. We looked for unauthorized accounts, trojan binaries, or any other backdoors. None were found.

Bob and I then decided to close some vulnerabilities that we knew would be attractive targets for the next attacker. Among the holes we closed were:

- rpc.cmsd – The calendar management services daemon, vulnerable to exploits (4). Because the business was not even using the service, we disabled it on each server by:
  1) editing the /etc/inetd.conf, using VI
  2) finding the line containing the following:
  100068/2-5 dgram rpc/udp wait root /usr/dt/bin/rpc.cmsd rpc.cmsd
  3) Commenting the line by inserting a # character at its beginning, and saving the changes back to /etc/inetd.conf
  4) Finding the relevant process ID (ps –ef | grep inet), then killing and restarting that process (kill –HUP <process ID>)
- sadmind – A systems administration utility, also not used by the company, but vulnerable to exploits (5). We disabled it on each server with the same process as above, but instead commenting out the line containing:
  100232/10 tli rpc/udp wait root /usr/sbin/sadmind sadmind

- rpc.ttdbserver – The tooltalk database server, also vulnerable to buffer overflow exploits (6). However, the business required this program to stay in operation, so we chose to patch the daemons with "Sun Patch 105802-05".

The next step for us was to install some reactive protection on the hosts, in other words, host intrusion detection (HID). A HID system watches activity on the host for suspicious behavior primarily by analyzing audit logs in real-time. A HID system also copies key parts of those logs to a centralized host, essentially creating a secure shadow log. Additionally, many of these systems can also monitor critical binaries or files for changes, checking every few seconds because any change might indicate a trojan.

I had already been investigating HID solutions as part of a security project; one of those solutions was Symantec Intruder Alert. Bob and I quickly installed Intruder Alert agents on the affected boxes, and configured the software to report any notable events back to us.

It was then time to perform a fresh vulnerability analysis of the systems using ISS Internet Scanner. The tool works by portscanning designated targets, analyzing what traffic it receives on the open ports found, and then cross-referencing that analysis with a thorough database of known exploits. The reports it provides are very useful, though not always accurate.

The vulnerability analysis did not find anything new, so the final step for us in this phase was to increase the sensitivity of the NID sensors. We reconfigured the sensors to not just alarm, but also fully log every packet in a session that matches the Mstream signature. This way, if it happens again, we will have better evidence in hand.

<u>Phase 5 – Recovery</u>
Returning the affected systems to duty was a relatively simple matter. Since nothing destructive was done, it was largely a matter for management to decide when they wanted to go back online.

The new ESM and HID systems continued to function long after the incident was closed. The sensors provided a steady stream of security information for me and Bob. We have continued monitoring the systems to this day.

<u>Phase 6 – Follow Up</u>
The following Monday, Bob and I met for a follow up meeting, and prepared a briefing for management.

It was unfortunate that we could not find the Mstream binaries, essentially the smoking guns in the case. We formulated two theories to explain this absence:

a) Our investigation, or other events, inadvertently triggered a self-destruct mechanism in the malicious code. Certain tools that may produce packets looking like Mstream (Tribal Flood Network 2000 for example) are known to have the ability to delete themselves if detected (2).

b) Or, the entire event was a false alarm, brought about by a NID system that was still relatively new and unoptimized.

Either way, Bob and I learned several things we could have done better, including:

- Preserve more evidence – Our incident handling inexperience at the time, coupled with an excitement to solve the case, led to us tainting the system earlier than was necessary. We should have moved all the systems to a private subnet, and sniffed the network for hidden time bombs or self-destruct triggers.
- Keep better evidence logs – Another area where our inexperience affected us was in the quality of logging our actions. We read the SANS guide recommendations, and started off with good intentions, but as the evening wore on the documentation ceased.
- Prepare for next time – Some missing supplies and information prolonged our time handling the incident. This experience taught us the need for a jump bag and the value of detailed documentation of the environment.
- Improve defenses – The incident also exposed several critical vulnerabilities in the environment, and opportunities for the company to improve its defenses. We documented these and developed plans to remedy them.

We presented our briefing and lessons learned to the management later that week. The company had been a quiet place and, although there was shock to learn it might have been compromised, the briefing was received well. While all the managers were concerned, the CIO was probably the most alarmed. He soon thereafter implemented sweeping changes that brought about:

- A Bigger Security Staff – The staff has doubled. Whereas we were only three before, the department has already grown to six in just a few months. By the end of the year we plan to have at least nine people fully dedicated to security. This includes Bob, who hopes to transfer in next month.
- Better Organizational Alignment – Before the incident, the department was two levels below the CIO. Now reporting directly to him, security benefits from a much greater authority within the organization.
- Bigger Budgets – Now that additional personnel are starting to arrive, there is finally funding for important projects like improved network and application security processes and reviews.
- Better Infrastructure – The security tools proved their worth during the incident. The NID system has nearly quintupled in size with nine sensors operating today. We decided to purchase ESM, and will begin formal deployment in a few weeks. In addition, we will soon also settle on a HID system, and deploy it on all the critical servers.
- Incident Handling Plan – Of course, most importantly, there is now a formal incident handling plan. The team is being formed and a comprehensive jump bag will be created. The team will even have a dedicated war room, complete with security servers, conference lines, and a large safe for evidence preservation.

Attachment A:

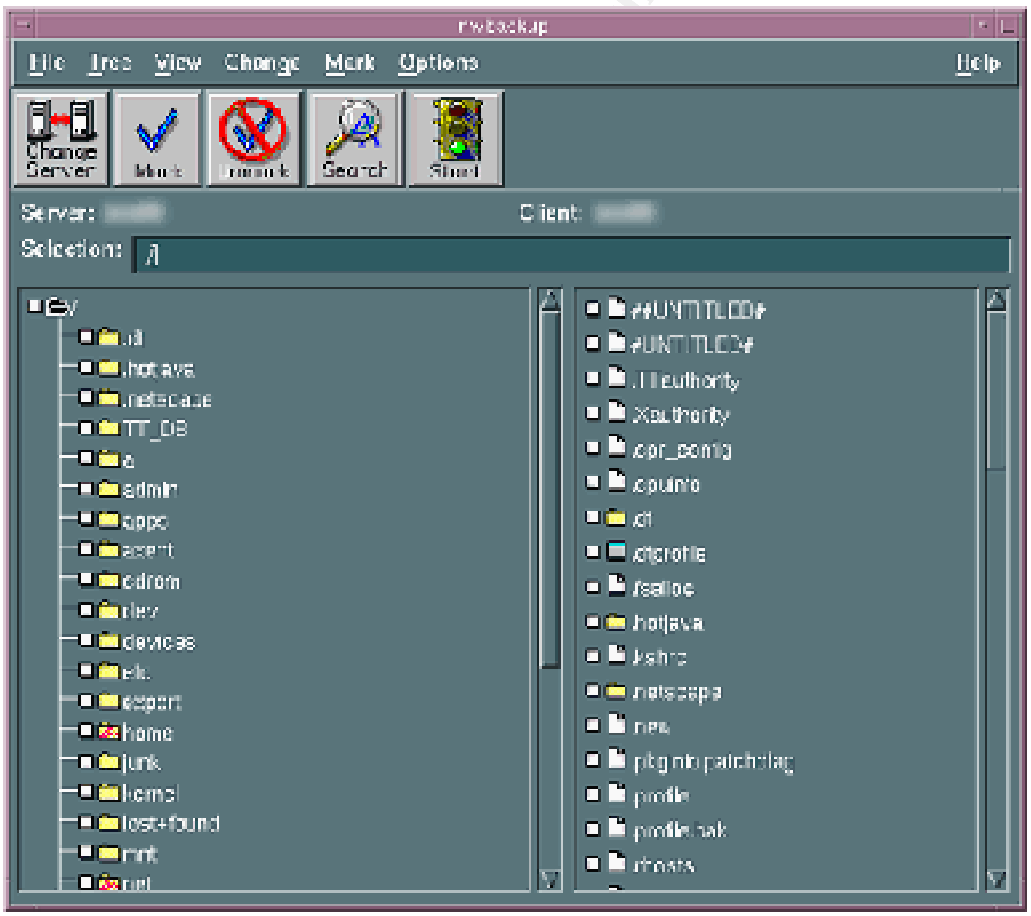
For at least one operating system involved in the incident, describe in detail the process used to back up the system. This should include descriptions of the hardware, commands used, and any problems that you ran into.

Our company uses Legato software to backup its servers. However, since our company only permits Legato certified operations personnel to use the software, Bob and I had to deputize one of those individuals into performing the system backups.

An interview with that operations person shortly afterwards documented:

The equipment used:

- Legato Networker v5.5.1
- Sun E450 server with a 100 tape ATL library jukebox, with DLT7000 drives
- Brand new 40gb DLT tapes



The essentials of the process used were:

1) The Computer Associates TNG software used to monitor the affected boxes was disabled. This was required to prevent dozens of alarms when the servers went offline.

2) The affected servers were then physically unplugged from the production network, but kept on a private backup network. This was done to remove all active users, and minimize the number of open or locked files.
3) The Legato software was started with command 'nwbackup'.
4) The affected servers were selected (one at a time) for backup, and then their desired filesystems and files were selected. In this case, all the local filesystems were selected. I cannot detail server names for reasons of confidentiality.
5) The 'Start' button was clicked to begin the backup process.
6) After the backups were complete, the used tapes were: removed from the tape jukebox; signed and sealed in Ziploc bags; then secured in a double-locked container located in Bob's office.

One problem encountered during backup was that one of the two web servers was not properly synchronized with its failover twin. This prevented us from getting a full backup of that server, and required us to settle for a differential backup.

Attachment B:

Describe in detail the chain of custody procedures used, any affirmations, and a listing of all evidence.

With no incident handling plan in place before the event, the correct procedures were not followed. A chain of custody is needed to guarantee the authenticity of any evidence collected.

The three key pieces of evidence in the incident were:

- ISS RealSecure NID logs – these were unfortunately destroyed by an unrelated system crash a few weeks later.
- Event Log – during our handling of the incident that day we did document most of our key actions:
  All of the following events occurred on October 27th 2000:
  - 01:29 ISS RealSecure Extranet sensor detects Mstream master packets from xxx.xxx.xxx.xxx to xxx.xxx.xxx.xxx.
  - 01:29 ISS RealSecure Internet sensor detects the same.
  - 09:30 Ken Gallo receives report of above activity.
  - 09:45 Informed direct manager of situation.
  - 09:50 Contacted 'Bob' from Unix systems administration team, deputized him for duration of incident.
  - 12:30 Received permissions to take affected servers down for backup and quick analysis. Called 'Jim' in Operations to begin backup. Opened new package of DLT backup tapes for exclusive use in this incident.
  - 12:51 Computer Associates TNG monitoring software is disabled on affected servers to allow quiet backup.
  - 12:55 Pulled web server #1 out of Cisco Local Director load balancing.
  - 13:05 Began full backup of web server #1 and related content server.
  - 13:15 Commandeered spare Sun E420R server, identical hardware to existing web servers, in case forensic analysis is required.
  - 13:30 First status report to upper management
  - 14:10 Web server #1 and content server backup done, installed Symantec ESM agents, began "Phase 3:c strict" audit. Removed used backup tapes from jukebox and secured in double-locked container located in Bob's office.
  - 15:00 Second status report to upper management.
  - 16:35 First ESM audits finished. No Mstream binaries found.
  - 17:10 Installed Symantec Intruder Alert agent on web server #1.
  - 17:40 Web server #1 put back in Cisco Local Director load balancing, web server #2 pulled out.
  - 17:55 Began backup of web server #2 and related content server. Only able to perform differential backup.
  - 18:00 Third status report to upper management.
  - 18:40 Web server #2 and content server backup done, installed Symantec ESM agents, began "Phase 3:c strict" audit. Removed used backup

tapes from jukebox and secured in double-locked container located in Bob's office.

18:50 Began full backup of other affected servers.

21:00 Fourth status report to upper management.

21:30 Second ESM audits finished. No Mstream binaries found.

After 21:30 Bob and I fatigued of documentation and stopped recording activities. This is a common mistake of incident handlers, and we should have kept better logs for the duration of the incident.

- Backup tapes – the final and most critical part of the evidence, the tapes are still preserved, signed and sealed, in a double-locked container in Bob's office to this day.

Attachment C:

REFERENCES:

1) "Incident Handling, Step-by-Step". SANS Institute 1998, pages 3-24.
2) "Computer and Network Hacker Exploits Step-by-Step, Parts 1 & 2". SANS GIAC Institute 2000, pages 311-315.
3) "ISS RealSecure 5.0 Signatures", Internet Security Systems 2000, pages 159-161.
4) "Multiple Vendor rpc.cmsd Buffer Overflow Vulnerability", SecurityFocus.com, http://www.securityfocus.com/bid/524
5) "Solaris sadmind Buffer Overflow Vulnerability", SecurityFocus.com, CVE-1999-0977, http://www.securityfocus.com/bid/866
6) "Multiple Vendor ToolTalk RPC Service Overflow Vulnerability", SecurityFocus.com, CVE-1999-0003, http://www.securityfocus.com/bid/122

Last Updated: July 28th, 2017

Upcoming SANS Penetration Testing

Security Awareness Summit & Training 2017	**Nashville, TN**	**Jul 31, 2017 - Aug 09, 2017**	**Live Event**
SANS San Antonio 2017	**San Antonio, TX**	**Aug 06, 2017 - Aug 11, 2017**	**Live Event**
SANS Boston 2017	**Boston, MA**	**Aug 07, 2017 - Aug 12, 2017**	**Live Event**
SANS Prague 2017	**Prague, Czech Republic**	**Aug 07, 2017 - Aug 12, 2017**	**Live Event**
SANS Hyderabad 2017	**Hyderabad, India**	**Aug 07, 2017 - Aug 12, 2017**	**Live Event**
Mentor Session - SEC542	**Des Moines, IA**	**Aug 14, 2017 - Sep 13, 2017**	**Mentor**
SANS Salt Lake City 2017	**Salt Lake City, UT**	**Aug 14, 2017 - Aug 19, 2017**	**Live Event**
SANS New York City 2017	**New York City, NY**	**Aug 14, 2017 - Aug 19, 2017**	**Live Event**
SANS Chicago 2017	**Chicago, IL**	**Aug 21, 2017 - Aug 26, 2017**	**Live Event**
Community SANS Memphis SEC504	**Memphis, TN**	**Aug 21, 2017 - Aug 26, 2017**	**Community SANS**
Virginia Beach 2017 - SEC560: Network Penetration Testing and Ethical Hacking	**Virginia Beach, VA**	**Aug 21, 2017 - Aug 26, 2017**	**vLive**
SANS Adelaide 2017	**Adelaide, Australia**	**Aug 21, 2017 - Aug 26, 2017**	**Live Event**
SANS Virginia Beach 2017	**Virginia Beach, VA**	**Aug 21, 2017 - Sep 01, 2017**	**Live Event**
Mentor Session AW - SEC504	**Milwaukee, WI**	**Aug 23, 2017 - Sep 29, 2017**	**Mentor**
Mentor Session AW - SEC504	**New York, NY**	**Aug 24, 2017 - Sep 08, 2017**	**Mentor**
Mentor Session - SEC504	**Denver, CO**	**Aug 29, 2017 - Oct 10, 2017**	**Mentor**
SANS San Francisco Fall 2017	**San Francisco, CA**	**Sep 05, 2017 - Sep 10, 2017**	**Live Event**
SANS vLive - SEC504: Hacker Tools, Techniques, Exploits and Incident Handling	**SEC504 - 201709,**	**Sep 05, 2017 - Oct 12, 2017**	**vLive**
SANS Tampa - Clearwater 2017	**Clearwater, FL**	**Sep 05, 2017 - Sep 10, 2017**	**Live Event**
SANS Network Security 2017	**Las Vegas, NV**	**Sep 10, 2017 - Sep 17, 2017**	**Live Event**
SANS Dublin 2017	**Dublin, Ireland**	**Sep 11, 2017 - Sep 16, 2017**	**Live Event**
Mentor AW - SEC504	**Santa Clara, CA**	**Sep 11, 2017 - Sep 22, 2017**	**Mentor**
Community SANS Columbia SEC560	**Columbia, MD**	**Sep 11, 2017 - Sep 16, 2017**	**Community SANS**
Community SANS Toronto SEC542	**Toronto, ON**	**Sep 11, 2017 - Sep 16, 2017**	**Community SANS**
Mentor Session - SEC560	**Dallas, TX**	**Sep 13, 2017 - Nov 15, 2017**	**Mentor**
Community SANS Madrid SEC560 (in Spanish)	**Madrid, Spain**	**Sep 18, 2017 - Sep 23, 2017**	**Community SANS**
Mentor Session - SEC504	**Arlington, VA**	**Sep 20, 2017 - Nov 01, 2017**	**Mentor**
Mentor Session - SEC560	**Manchester, NH**	**Sep 21, 2017 - Nov 02, 2017**	**Mentor**
Community SANS Columbia SEC504	**Columbia, MD**	**Sep 25, 2017 - Sep 30, 2017**	**Community SANS**
SANS Baltimore Fall 2017	**Baltimore, MD**	**Sep 25, 2017 - Sep 30, 2017**	**Live Event**
SANS London September 2017	**London, United Kingdom**	**Sep 25, 2017 - Sep 30, 2017**	**Live Event**