Global Information Assurance Certification Paper

An Incident, or a Tragedy of Errors?

Fred Berryman
March 16, 2001

I was recently asked to assist in the investigation of a large site that had been reported as being compromised. The site contained an equal mix of Microsoft and Unix systems. I used the word "reported," because we were never able to produce any clear, factual evidence that a compromise had occurred. The SANS definition of an incident is:

> *"An adverse event in an information system, and/or network, or the threat of the occurrence of such an event."* [i]

After reading this paper, the reader can decide if an incident had actually occurred, or if the events had been caused due to administration issues, which led to an uncontrolled environment. Regardless of the answer, this paper should be a warning to enter a site with caution, and that any remedial actions should be preceeded by a reasonable period of investigation.

The six stages of incident handling as defined by SANS will be covered: preparation, identification, containment, eradication, recovery and follow up or lessons learned. I will document the actions taken at each stage and point out the mistakes that were made along the way. Keep in mind, that incident handlers do not wish to cause further harm on a network that already has problems.

Preparation:

My partner, Alice and I have been working on the routers, switches and applications in this particular area of the company's network for many years, but we are both new at incident handling. Alice knows routers and hubs very well and I have some experience with Intrusion detection systems and firewalls. Neither of us is proficient at Unix or NT, but we know enough to be careful. We were well aware of the need for creating backups of the potentially compromised systems to preserve both user and forensic data, but getting the system secured and operational was our main goal.

Most of what we knew about incident handling was learned from attending SANS conferences and completing a couple of SANS courses. We had some tools available including sniffers and intrusion detection software. Some Sans courses we had completed consisted of Level1, Firewalls administration and an assortment of other classes at Sans conferences in the past two years. We configure and maintain firewalls, routers and intrusion detection devices for several locations.

What we did not have was a backup device that could be used to store all the potential forensic data. Additionally, there were complications with the standard backup system for the environment that I will discuss later within this document.

Our emergency communications plan was simple, we (Alice and I) needed to go on site to perform a preliminary evaluation and report our findings back to our team. PGP was utilized to encrypt Email conversations among our team members during the incident. We used cell phones to keep our conversations off the local switchboard. Since this turned into a rather lengthy visit, we soon learned that Stephen Northcut's advice to carry spare cell phone batteries is a good idea.

There is a corporate level security person who is our liaison with law enforcement and the media. Our policy states that all communications with either law enforcement or the media must be funneled through this individual.

The incident occurred at a site that we do not normally visit though we knew most of the individuals involved from prior work relationships. Escalation of our concerns and our discoveries to management was difficult due to many of them being on vacation for the holidays. However, we suggested that the local personnel keep their individual managers informed of our activities from the beginning, whether they were on vacation or not.

Identification:

The incident started when Bob, the site NT administrator, reported strange activity that had been occurring on his Windows systems. Alice and I met with Bob for lunch to gather additional information in a neutral setting. Bob described the following incidents:

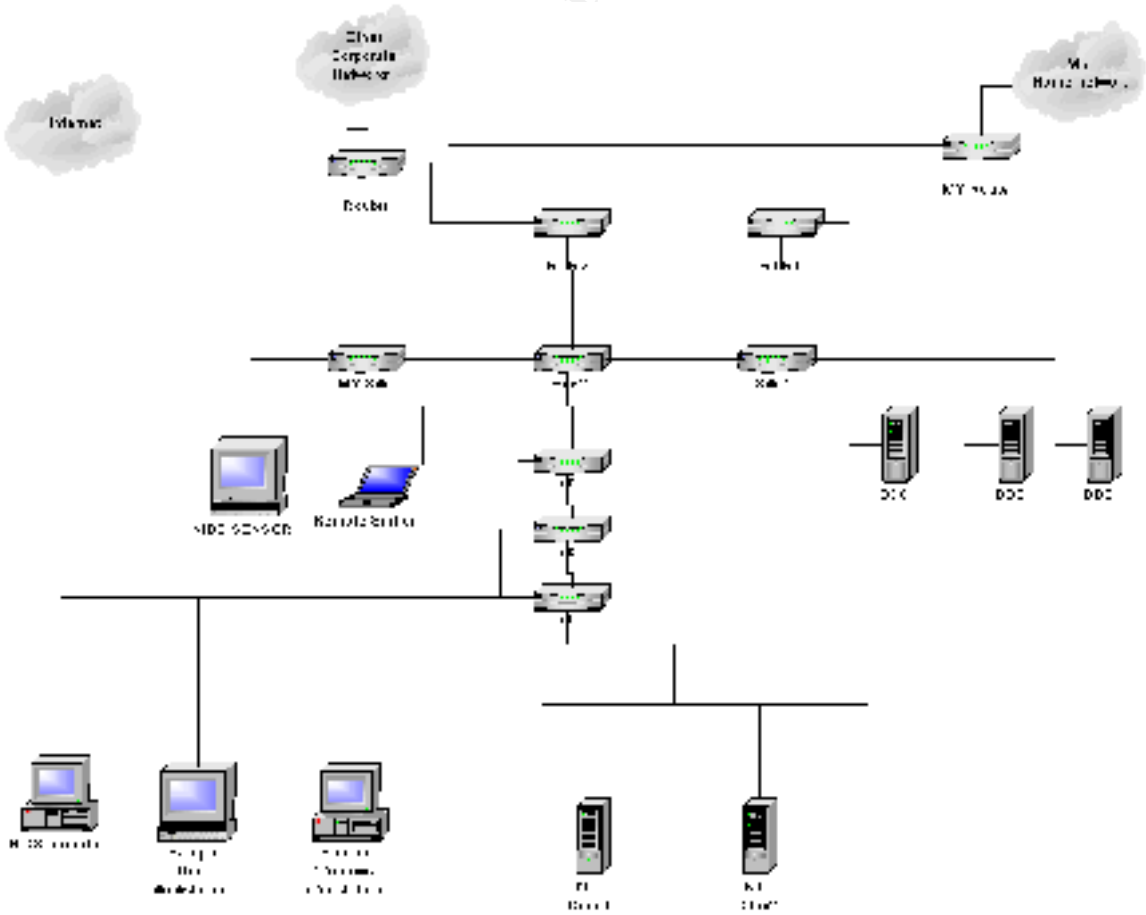
1. Entries in the NT Security log showed that users, which were on vacation or out of the office, were logging into systems.
2. These logins were not only gaining access to the user's machine, but were also escalating their privileges to admin on the NT servers.
3. Entire directories were being removed from the Citrix Metaframe [ii] servers.
4. The tape backup system had been rendered inoperable. This meant that many of the systems, which included systems with large disc arrays, could not be backed up.

The NT logins were initially the biggest concern to Bob. He apparently spends a great deal of time monitoring his systems and he occasionally detected activity that he could not explain. Monitoring logs is admirable, but the suspicious activity had been occurring for some time before a report was made.

Yes, there was some suspicious file activity at the site. All the users kept their data on and ran their applications from the Citrix servers. Some were experienced and some were new enough that they may have made some mistakes that caused their data to disappear. Learning to tell the difference could make a big difference in your career path.

Bob contacted the vendor of the backup system and they told him there was nothing he could do but reload the system which meant losing the directory and hence the historical data on the backup arrays. We later accepted the situation and reloaded the backup system software and performed full backups of all the systems, before performing other activities that could have alerted the hacker(s), if they actually existed.

We had access to the network via a dedicated T1 that connects their site to ours and we had full control of that portion of the network, so we had an excellent view from a distance. Here is the network diagram we drew while on site.



There were things we would like to change with the layout, but we were there to handle an incident, not to redesign their network. The only way to the other corporate networks was via RTR2. The servers and workstations were divided between two class C networks, and then connected again at the hubs. The switches and the sniffer in the upper left corner of the diagram belong to our group. They allowed us to log all traffic from both subnets. We configured spanning ports on each of their switches and ran those over to our switch. At our switch, we configured a spanning port to run both of these ports to a sniffer port. The switches they were using were old and incapable of giving us this functionality, so we needed to install two new switches.

We also connected an intrusion detection sensor to the switch; since it was the only place we could see all traffic at the site. The actual configuration and deployment of the Network Intrusion Detection System (NIDS) sensor is described in the next section.

We already had a Host Based Intrusion Detection System (HIDS) management station running in our location and we wanted the ability to run scans remotely, so we only needed to load HIDS agents on the suspected servers.

We configured our syslog server to collect data from their machines and configured their *syslog.conf* files to point to our server. This is so the hacker could not destroy log data to cover his or her tracks and remain covert. We did not encrypt the syslog data since it only travels over our own circuits, making it difficult for a hacker to intercept.

We scanned all the major servers with HIDS that could be loaded remotely and gathered enough log data from various machines to make an initial assessment. We found a large number of basic security configuration errors. Logging was not enabled on most machines and where it was enabled, they weren't logging enough data to help us much. Trusted system was not configured on the Unix machines, which means the password were stored in the /etc/password file, making the hackers job just a little easier. Trusted systems splits the login id's from the passwords by deploying the /etc/shadow file. Normal users had rights to things that only the administrator should have [iii]. Instead of putting the backup operator in the appropriate user group, he or she was given the admin password instead. If the lack of host security wasn't enough, the router connecting them to the outside world was Access Control List (ACL) free.

The Citrix Metaframe and the Windows domain servers were the core of the network. Users kept all their applications and data files, including email, on these two servers. The main reason for the Citrix system was to reduce the number of workstations required at each desk. Unix machines ran Windows application, or at least the Citrix application made it look that way. Taking these essential servers down to reload the operation system was not something any of us wanted to rush into.

As I stated before, we wanted to do as much research as possible from offsite to prevent alerting any potential intruder, assuming this could be the work of an insider. We decided to scan as many servers as possible from our site, across the network. Normally, loading an HIDS agent is as simple as inserting a CD and answering a few basic questions. In this case, we had to load the CD at our site and advertise it to the site of the incident, then mount that CD from the remote machine. There are many little syntax anomalies that can delay incident handling. This is just one that caused me some grief. Here are the commands to perform this task. Due to corporate security policies, I must remain a little vague as to the actual commands, but you should get the idea.

On the local machine, load the CD and issue these commands

mount /dev/dsk/c1t2d0 cdrom,

exportfs –i –o ro /cdrom

an alternate method is to edit the /etc/exports file, but I found exportfs to be easier

On the remote machine

mount the remote share with this command

/etc/mount myhp:/cdrom cdrom (this assumes you know the full path of myhp)

or use SAM, nfs mount remote

Our HIDS is designed to be auto loaded and executed. The syntax of the file names forces the use of the * wildcard to remote execute the command.

cd /cdrom/HP/HPUX/PARISC/ESM50,

*./HIDS** The actual filename on the cd is ESM50;1, the * is used here to get past the semicolon in the filename.

When you are finished, get out of cdrom directory and execute this command

/etc/umount cdrom

Also, this particular HIDS requires that the client machine must be able to ping the manager by name, so if the manager is not in DNS, add it to /etc/hosts on the client and add the client to the /etc/host file on the manager.

Containment:

Armed with an understanding of the importance of the site and the vagueness of the incident, we entered the site with caution. We kept a low profile by entering the site at night when most of the staff was gone and we worked in the server room, which is out of site and locked. We installed a remote sniffer to monitor all traffic at the site and we put in router access lists to prevent unnecessary access from the outside. We deployed the NIDS sensor mentioned above, sending its reports to a remote console offsite.

We found some modems configured to accept calls from the outside and turned them off. An interesting point about the modems is that no one knew what the modem was for or why that particular machine needed one. I think the phone companies like

sites like this and so do the bad guys. There were modems that were only needed for testing during the day, so we instructed the site administrators to turn them off at night. We changed all the administrative passwords on domain servers and the Citrix boxes (several times during our visit).

We monitored sniffer traces and event logs for days. When Bob, the system administrator reported illicit user logins, we studied sniffer logs to see how the potential hacker was getting in. As it turned out, most of the logins were explainable. Many users had clients configured to pull down mail, even when they were on vacation. These application logins look amazingly like a user trying to gain access.

Some confusion arose from the amount of netbios traffic involved when a user on a UNIX box logs into a Citrix Metaframe database. The Citrix is configured to use the NT primary domain controller (PDC) and backup domain controllers (BDC's) for user access control. Packets are sent from the user, to the Citrix, to the PDC or maybe to a BDC, then to the PDC. A sniffer trace of this activity was confusing and difficult to justify, since we did not have extensive NT experience, but it was an education in itself.

When the NT event log implies that a user has requested to have his rights escalated in the domain, is it really just showing what he could do, and not what he requested or received permission to do? This is the point that was bothering and confusing Bob. It looked like ordinary users were requesting if not gaining access to things in the domain, which they had no reason to have. He also reported suspicious admin activity that was actually normal traffic. For example, the following is an excerpt from such an event log that contained thousands of these same entries. We believe this was the domain controller doing some standard file maintenance, but our lack of experience left us a little unsure.

```
11/xx/00,10:04:39 AM,8,4,578,Security, NTNET\Administrator,,PROLIANT,Security
124 4058668356 Administrator  NTNET (0x0,0x2072E) - - - SeSecurityPrivilege
11/xx/00,10:04:39 AM,8,4,578,Security, NTNET\Administrator,,PROLIANT,Security
124 2160957760 Administrator  NTNET (0x0,0x2072E) - - - SeSecurityPrivilege
11/xx/00,10:04:39 AM,8,4,578,Security, NTNET\Administrator,,PROLIANT,Security
136 4058668356 Administrator  NTNET (0x0,0x2072E) - - - SeTakeOwnershipPrivilege
11/xx/00,10:04:39 AM,8,4,578,Security, NTNET\Administrator,,PROLIANT,Security
136 2160957760 Administrator  NTNET (0x0,0x2072E) - - - SeTakeOwnershipPrivilege
11/xx/00,10:04:39 AM,8,4,578,Security, NTNET\Administrator,,PROLIANT,Security
128 4058668356 Administrator  NTNET (0x0,0x2072E) - - - SeSecurityPrivilege
                                                       SeTakeOwnershipPrivilege
```

We had to admit the following log entries had us guessing for a while. Rick is the UNIX sysadmin, but he was on vacation when we saw thousands of these entries within a

few minutes of each other. As it turned out, Rick had accessed the system remotely to do some backups.

```
11/xx/00 9:46:41 PM 8 4 578 Security NTNET\Rick
LPSERV Security 672 2153902720 SYSTEM NT AUTHORITY (0x0 0x3E7)
RICK NTNET (0x0 0x6B8BD) SeTakeOwnershipPrivilege
11/xx/00 9:46:41 PM 8 4 578 Security NTNET\Rick
LPSERV Security 1284 4262976944 SYSTEM NT AUTHORITY (0x0 0x3E7)
RICK NTNET (0x0 0x6B8BD) SeTakeOwnershipPrivilege
```

The most disturbing NT login issue was when the event logs indicated users logging off without a corresponding login. This sounded suspicious and we never really proved how it happens, but we did document instances where BOB created the exact log condition when he logged out on occasion and we could not locate his login. My suspicion is that the login occurs on one domain controller, which actually checks in with the primary domain controller, then the logoff is handled by yet another domain controller.

There were times when the event logs would fill up with repeated logins from users that may not have been at their computers. For example, Rick was recorded by these entries, thousands of times in a few hours. I think this was just before a reboot and probably not worth mentioning, but we had to check it out.

```
11/xx/00,9:45:08 PM,8,4,578,Security, NTNET\ rick,, LPSERV,Security 988
4262976944 SYSTEM NT AUTHORITY (0x0,0x3E7) RICK  NTNET (0x0,0x6B8BD)
SeTakeOwnershipPrivilege
11/xx/00,9:45:08 PM,8,4,578,Security, NTNET\rick,, LPSERV,Security 988
2153902720 SYSTEM NT AUTHORITY (0x0,0x3E7) RICK  NTNET (0x0,0x6B8BD)
SeTakeOwnershipPrivilege
```

We realized that much of the suspicious activity might be malicious but that we were not prepared to make that determination. We asked for an NT specialist to join us. He confirmed our conclusion that much of the activity was normal and that the systems were wide open, allowing normal users access to things they didn't need, which they frequently used, making the network traffic and event logs look suspicious.

He noticed that the NT passwords were available in the NT recovery directory and told us that making an Emergency Repair Disk (ERD) on NT will change the permissions on the SAM database in that directory. A solution would be to run syskey to encrypt the SAM database, but we were hesitant to do this in the middle of the investigation. We simply changed the permissions on the recovery directories and asked the administrator to convert to Syskey as soon as possible.

There were stories of backdoors on servers and other such rumors that were never validated, but the suspicion was real. We never saw a single packet entering the site that was not warranted or explainable. There were strange things reported during our investigation that were apparently the result of the way the system was administered. For example, all users started receiving junk files one morning that had the extension mmf. Bob had read something about a Cult web site and recalled that they used this file extension. He became very excited at the prospect of wrapping this thing up. The files appeared, and then disappeared from desktops. We found no cult web sites or malicious software that distributed files with mmf extensions, so we dismissed the cult story.

The mysterious traffic made me curious enough to want to see more of what goes on at the site, so I installed a tool called Session Wall from Abirnet,[iv] which monitors traffic and provides a great GUI interface report of the traffic. You can see what each and every user is up to. I monitored the traffic for a 24-hour period. The final result was that there was a small amount of time being wasted on non-business related web browsing, but nothing malicious was apparently going on.

We established a syslog server offsite and sent syslog data from the main servers at the compromised site to our own computer. This prevented any potential hacker from modifying or deleting them. Since we already had an offsite syslog server running, all we had to do was to tell the server's *inetd.conf* file to start *syslogd* with the –r option and configure the local *syslog.conf* files to send certain events to our server. All the machines involved already had udp 514 (syslog standard port) configured in /etc/services.

The site does not deploy TcpWrappers, so we used other means to control access from tcp services. We had the sysadmins configure the /etc/ftpusers files to control ftp access and we configured *inetd.sec* to control telnet access. We shut off unneeded ports, protocols and services like IPX, chargen, and echo. I actually started a echo-chargen storm and panicked a little when it didn't want to die. We gathered forensic data with commands like ls, who and netstat. We collected files like /etc/services, */etc/syslog.conf, /etc/hosts.equiv, /etc/profiles, /etc/groups, /var/adm/sulog* and saved the results from occasional *netstat –p tcp, last and lastb* commands. Nothing was learned from these activities except that the systems were wide open and needed some serious work. Trojan system commands could have been fooling us if the systems had been root-kitted, but lack of training and experience prevented us from detecting any such file tampering.

Finally, there was one event that we never explained away, which remains our only real, observable event that we thought justified the eradication steps taken below. During the investigation, we observed users getting a *winlogin* screen at a point where they never had to login before. It appeared to be a Trojan horse that may have been grabbing user logins. Once each user logged in at this prompt, he never saw it again. Each user got the request for login and password one time. This was disturbing and had

to be dealt with immediately. A lesson learned here was to take a screen shot when something unexpected pops up.

Eradication:

This is the stage where we will all take our own paths. Observing strange activity is easy, while determining the source of the activity is a bit more difficult. Alice read megabytes of sniffer traces, trying to discover how a hacker may have entered the site or system to cause each reported event. Most if not all the events were explained away as normal for this environment. The volume of netbios data traversing this network when a user logs in is amazing. The user requests access to the Citrix server, which asks the NT domain for authentication. If the request hits a BDC, the BDC asks certain things of the PDC and certain log entries get made that look suspicious, but are actually innocuous.

The *winlogin* screen was all we needed to declare that one real event makes up for a lot of false positives. We proceeded as though there was at least one bad guy on the inside who was potentially sending sensitive data to the outside via IRC or some other harmless looking service.

Before making any network changes, we decided to rebuild the corrupted backup system and make a clean set of backups. We lost potentially useful forensics data, but what could we do? The vendor already told us the data was lost.

We drafted a letter to local management, summarizing some of our findings and requesting that some machines be rebuilt and other steps to start the cleanup process be taken immediately. We requested these actions:

1. Make a complete backup of all systems.
2. Change all user logins and demand hard to guess passwords.
3. Change NT domain controller passwords.
4. Change admin on local servers to something other than the one used on the PDC.
5. Turn off all workstations not being used to reduce network traffic.
6. Rebuild the Citrix systems from scratch.
7. Rebuild the PDC and BDC's.

Users started cleaning up their Citrix home directories and found that they could not delete some of the files. As it turned out, the junk email had been distributed to all users due to a directory move on the Citrix servers and the files that could not be deleted were due to the hidden or system flags being set on some old files. Some machines were reported to be out of disk space "all of a sudden" but when we checked them out, the owners stated that they had not checked the free space on their systems for many months

and all the files on those systems were justified. Again, it is easy to jump to conclusions when tension is high. Be slow to accept any report as malicious activity until you confirm it yourself.

The Sans Incident Handling and Malicious Software course talked about the use of IRC by hackers to setup covert channels of communication, so we were alerted when we saw IRC packets leaving the site. We never proved that Speakeasy or Trumpet was being used, but we blocked IRC from the site anyway. v

One of the Domain controllers had crashed just prior to the report of the incident. We had Bob rebuild the box and it ran for a while, and then crashed again. The error said the machine couldn't see the ethernet card. We suspected a hardware problem, while Bob was convinced it was a hacker (we never saw his logic). We had a new box shipped in for them to use while they fixed theirs. The "hacker versus hardware" analysis became a point of disagreement between our team and local administrators to the point that communication was affected between the groups. We tried to convince Bob that we were there to help, not to throw blame on him or anyone else. Our job was to observe, document and recommend, not to prosecute. Bob found another position, and left the company. We were able to finish the incident by working with Rick and other local staff, who were extremely able and willing to help.

Recovery:

Since we suggested rebuilding all the servers, not just one system, the recovery was done in stages. Site support functions had to remain online at all times, so a workaround had to be deployed. They installed standalone PC's with all the applications required to service customers and they used modems instead of network access. This wasn't done willingly, but after a meeting with local staff, where Alice and I explained the consequences, they followed our advice.

Rick put on the NT hat along with his Unix hat. With Rick's help, we cleaned up many of the vulnerabilities reported by the HIDS. Alice continued working on the router ACL's to allow only traffic needed in and out of the site and only to and from specific addresses. Rick loaded security patches and removed the web of trust that had been allowed to grow beyond reason. He disabled remote access and control of the domain controllers and shared directories only when necessary. His cleanup is still in progress as of this writing.

Rick made a new PDC, moved everyone over to it and then rebuilt the BDC's. Once the domain controllers were rebuilt, the Citrix Terminal Servers were rebuilt one at a time. At this point, we realized that the two Terminal Servers that were supposed to be mirror images of one another, had little in common. This fact helped to explain some of the suspicious file activity in user directories. If a user was moved from one Terminal

Server to the other, his account and applications looked different, because of the differences in the two servers.

We removed all unnecessary modems and suggested a policy be written and distributed that required modems be disconnected when not being used. Policies statements were also issued to enforce the use of strong passwords and to enforce the rule of least privileges.

Follow-up / Lessons Learned

It is apparent to me that my jump bag is not complete. I found myself wanting tools that I did not have. The first thing that comes to mind is a program to run syslog on NT like SL4NT or Kiwi's syslog for NT. When I started looking at Unix machines, I found myself relying on site copies of potentially corrupted binaries. SANS recommends building a set of binaries stored offline. I have since begun to build a CD with these tools.

We spent the first week onsite without a copy of the NT resource kit, so there were things I had read about in the SANS courses that I could not do. We did not have our own tape backup system, but I feel we performed the backups soon enough to be useful, even though we were never convinced that there was an intrusion. We employed Microsoft Office tools like Word to keep notes and drawing tools like Visio to draw diagrams. We had NIDS and HIDS tools available and used them extensively, but we did not have a copy of the SANS Intrusion Detection Step by Step guide and the forms included in the back of the guide. Having a check-off list for keeping track of what was done on each machine would have been useful.

As for forensic data, we still have the sniffer logs, system log files, snapshots of configuration files and all the HIDS scans that were performed. If the problem returns, we will not have to start over. We do not have a strict requirement for gathering forensic data because our main concern is system restoration.

It is apparent to me now that it is no simple task to find or prove the existence of hacker tools like BO2K, root kits and tunneling software. Sniffer traces are great, but they will not detect BO2K wrapped around a copy of some innocent looking executable like a birthday card or word.exe with a tool like SaranWrap. I must take the time to analyze network traffic on a clean system, especially when Netbios is used, so I can more easily detect hostile traffic during an incident.

For follow-up, we Alice and I were asked to spend 4 hours per week, working with the administrators to answer questions and assist in the cleanup process. The local staff had some problems finding patches and software that we and the HIDS had recommended. Alice received a few calls a week at first for router ACL changes to meet

customer needs. We left the sniffer and NIDS in place for one month and looked at the logs daily for the first week, then only occasionally after that. Local management indicated that the systems are running more smoothly and that the impact on operations was less than we had feared.

In conclusion, not all events are incidents, and malicious hackers do not cause all enents. Poor administration habits could likely be the main problem and it may take years of incident handling to quickly discern the difference.

---

[i] Stephen Northcut, SANS GIAC Advanced Incident Handling and Hacker Exploits, IH_emergentcyresponse.pdf, pp4, 2000

[ii] Announcing Citrix MetaFram XP, Citrix Home Page http://citrix.com, March 16, 2001.

[iii] Drew Heywood, Windows NT Server 4, Second Edition, New Riders, 1998, pp 204-258.

[iv] Abirnet now CAI, http://ca.com/solutions/enterprise/etrust/intrusion_detection/product_info/sw3_whitepaper.doc, 3/26/2001

[v] Eric Cole & Ed Skoudis, SANS GIAC Advanced Incident Handling and Hacker Exploits, BO2kRootkits.pdf, 2000