Global Information Assurance Certification Paper

Anatomy of an Insider Attack

Pierre J. Tarver
January 16, 2005
GSEC Practical Ver 1.4

Many security professionals have expressed concern over the so called “insider threat”. According to the Secret Service "The insider poses the greatest threat because they know where the most critical information is kept and how to bypass the safeguards on the system”.[1] The subject of this paper is an actual insider attack. The paper will first attempt a definition of terms. Next, it will provide background information on the company, the network in question, and the personnel. The paper will then discuss how the incident was discovered and what attempts were made to identify the perpetrator. Finally steps taken to prevent the incident from occurring again will be analyzed. The company and all identities will remain anonymous. Any software mentioned is not necessarily an endorsement of one particular software package over another (there are many available), rather, it is simply a statement of a particular package the company chose to employ. It is hoped that this paper will provide some insight to others, so that they may mitigate the risks from an insider attack.

Before a discussion of an insider attack can begin we must define what is meant by the term "insider", Peter G. Neumann, of the SRI Computer Science Lab provides a workable definition. He defines insiders as" relative to a particular computational framework, insiders are users who have been authenticated to operate within that framework".[2] Specifically, one would argue that it is a user that violates the degree of trust afforded them to operate on a given computer system. For example, a bank teller is given trust to access the records of customers. If the teller in turn sold the information to a third party, without the bank and the customers' knowledge, this would be considered a violation of the implicit trust. The teller has the trust to access the records, not to sell them. It is this violation of trust that defines an insider attack. This paper will now describe the Company itself, and its security policy prior to the incident.

Company Background:
The company in question is a large aerospace company. Clients include both the United States government, and private corporations. The company deals primarily with sensitive information. The company has external networks connected to the Internet and internal networks that have no Internet connections. The incident occurred on an internal network.

Staff: Three full time system administrators (Admin) maintain the internal network. In addition there is one Information Systems Security Officer (ISSO) who is responsible for advising on computer security policy and issues. The Admin and the ISSO both report to a Contract Program Security Officer (CPSO) who has overall authority for all security and configuration issues. The CPSO has two assistants (ACPSO's) who assist them. In addition, approximately one hundred and eighty support staff and engineers have active accounts on the network.

Network Background: The network in question consisted of one hundred and fifty four client computers, connected to three central Servers. One hundred and four of the client computers run Windows NT 4.0 as the operating system. The remaining computers are Sun and Digital Alphas running Unix. The network is spread out between eight rooms in three different buildings. The connectivity between the buildings is accomplished by dedicated fiber optic cable. All of the machines contain removable hard drives. When drives are not in use, they are stored in locked safes. Users do not sign the drives out, but rather remove them once the safe is opened at the beginning of the day. The safes are opened and closed by a member of the security staff who signs a sheet denoting the opening and closing times of the safe.

Security Policy: Because of the sensitivity of the work conducted by the company, strict regulations for conduct are enforced. All personnel are subject to background investigations by the Federal government. Users also agree to a polygraph by the United States government that may be given at any time. The entire Program is inspected annually by the government. Any deficiencies must be corrected within 30 days. In addition, all users must attend an annual briefing covering computer security and regulations. Among the instructions given to the employees are the following:

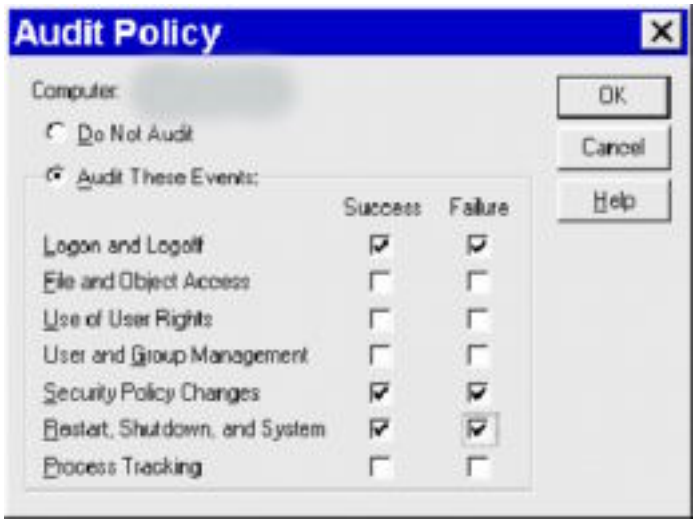
- Users may not conduct hardware changes. All hardware configurations are documented and only the Admin are allowed to perform hardware maintenance.

- Users may not install software. All media must be brought to security on entry into the Program area. Security documents the files, conducts virus scans of the media prior to installation, and then installs the software.

In addition, all Windows based machines run Anti-virus software, which has it's virus definitions updated weekly.

Network Security Audits

The systems are audited weekly by a system administrator and monthly by a systems security officer. The specific events audited are successful and failed User Logons, Security Policy Changes, and System Shutdowns. The following graphic (1a) shows the audit policy found on the Windows NT machines.



(Graphic 1a)

Audits on the Unix machines are audited similarly, with the addition of su to root also being audited. This is done using a customized shell script. Root privileges, and Administrator rights on the Windows machines are kept to a minimum. Any user who requires administrative privileges must have the approval of Management. In addition, a special brief is required which defines the responsibilities of privileged users.

Account Policy

Each user is given a unique id and password. Users are told to never share or disclose their passwords, or keep them written down. (For example, on post- it notes under their keyboards, on monitors, etc.) Passwords expire every 180 days, and must be at least 6 characters in length (combination of upper and lower case, and numbers and/or special characters). A password cracking utility is run annually to test the strength of user passwords. Users can not recycle the last 5 passwords they have used. Users are locked out of the system after 5 bad logon attempts. Graphic (1b) illustrates the account policy on the Windows NT client machines

(Graphic 1b)

The Incident

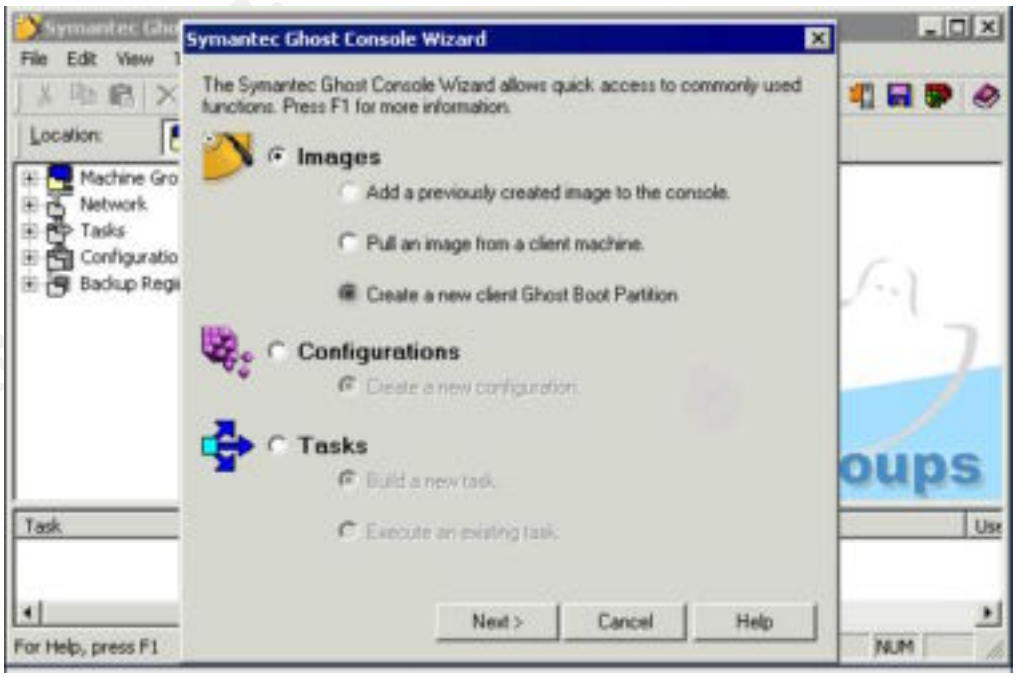
The incident was initiated when an admin rebooted a client machine while installing a software package. The admin noticed that LILO was loaded on the machine. LILO, which stands for "Linux Loader", is a boot utility that enables a user to boot Linux, and/or multiple Operating Systems, on a PC. The administrator discovered that the system had been configured to dual boot both Windows NT and Linux. There was no record of Linux being loaded in the Program Area. The admin notified the ISSO that there was a potential security violation. The ISSO requested the system be taken off of the network, and a meeting was convened with the Security Staff.

The first order of business was to try and determine if the incident was in fact a true security violation. It was here that a difference of opinion emerged. There was significant argument among the staff on whether to involve the government immediately, or try to compile a body of evidence first. Several questions were raised. For example, did a user simply installing Linux break any laws? What was the intention of the user? Was it done with benign intentions (a user simply wanting to use his Operating System of choice), or was there malicious intent (a user circumventing NT auditing, and running malicious software)? One staff member suggested that we ask everyone, if they had installed Linux, and simply ask them not to do such activities again. Another staff member suggested that we notify law enforcement immediately, and let them investigate. Several scenarios were argued. In the worst case scenario, there was the threat that the system was used for

espionage, or for theft of proprietary information. In the best case scenario, a user had simply ignored company security policy in an effort to increase productivity. In between lay the real possibility of having every employee subject to a government polygraph, straining already limited government resources and impacting company productivity. It was finally determined that the staff would have to attempt to ascertain the nature of the offense first. A decision was made to make a copy of the hard drive. The copied hard drive would be examined. This would be done to avoid risk of damaging potential evidence should law enforcement be needed. The staff would try to determine when the software was installed, who installed it, and why they installed it. Based on the findings, a decision would then be made on whether to involve law enforcement. An attempt would be made to keep productivity disruption to a minimum, but the original hard drive would be kept off of the network until the findings were complete. The investigation would also be kept secret from the staff. Users would be told that the hard drive had crashed and was being examined for repairs. Any data that was needed could be acquired from the back-up tapes. This caused significant argument among several of the staff. Some felt it unethical to "tell lies" to personnel. In the end it was felt that a "cover story" would be wise in the event that the worst case scenario played out. If there were serious malicious intent, it was felt that it would be better not to give the guilty party opportunity to take action that might hinder an investigation.

The Investigation Begins

The hard drive was "cloned" using *Symantec Norton Ghost Corporate Edition 7.5*.[3]. The software allows a user to make a working copy of the hard drive, and this copy would be examined to avoid contaminating the suspected drive. Ghost can be run in DOS mode, or in Windows mode. In Windows mode, Ghost provides a user-friendly interface, with a wizard to easily copy drives. Graphic 2a shows an example of the Wizard used to create a ghost image.



(Graphic 2a)

Once the drive was cloned, the original drive was locked in a safe in which only security personnel have access. A working log was created that documented times and activities, as well as naming the personnel conducting the action.

The initial inspection of the hard drive revealed that Linux had been installed approximately one month before. A user had evidently partitioned the Hard Drive without while preserving the data (most likely through the use of a utility such as Power

Quest's *Partition Magic*[4]). The Server logs were checked for the corresponding date and time, and only showed that the system had been shut down, and had remained down for approximately an hour and a half, before being restarted. The user identified in the log was noted, and it was noted that the same person who had shut down the system, had also restarted the system. Subsequent checks of the Server log had shown that on six separate occasions the same system had been shut down for times ranging from forty five minutes to an hour and a half, and then restarted. In all instances, the same user did it. This had not raised any alarms among the Security staff in the original audits, as this was not something that had especially stuck out as strange. Users often shut down systems to change hard drives. This alone did not prove much. Only that probably the same person had been shutting down the system, probably booting into Linux, and then booting into Windows when they were done. It would be necessary to actually look at the files on the Linux machine in order to attempt to discover the user's activity. This was done with the hope that the user's intentions could be discovered. The only problem was that the staff did not have an account on the Linux partition, much less root access. A discussion was conducted on the best way to try and crack into the system. In the end, it was decided to try the simplest things first. They would attempt to simply boot into single user mode, and see if they could simply assign a root password this way. Linux operates in different "run levels". These run levels specify the mode the system is to be run in, and are used because systems might be utilized in different ways. For example, if a system is not connected to any network, and has only one user, you may not want to have all of the communications processes running. Linux usually has 6 run levels. The following run levels are generally defined in Linux:

Run Level 0 — Halt. This halts or shuts down the machine.

Run Level 1 — Single-user mode. Usually used if a system is stand alone, and only used for a specific task.

Run Level 2 — not used (user-definable). Can be given a specific purpose by the user.

Run Level 3 — full multi-user mode. Starts network daemons, multiple user accounts, etc.

Run Level 4 — not used (user-definable). The same as run level 2.

Run Level 5 — full multi-user mode. Like run level 3, but with a GUI interface.

Run Level 6 — reboot. Restarts the machine.

When the system was booted, a LILO prompt appeared. Typing in the following command brings the user into single mode. (This may vary based on which distribution is being run).

LILO boot: linux -s

Once in single user mode, the root prompt appears. One needs to simply change the password, which is done using the following command:

```
# passwd
```

Once the password was changed, the machine was restarted in normal mode. (Run level 5 in the particular distribution that was being examined).

Findings

The system only had one user account named "user". An exhaustive review of the software on the system discovered that a program called *Sniffit* had been installed. The software is a *protocol analyzer* or *sniffer* program. Sniffer programs allow network traffic to be "sniffed" or monitored. Although originally designed to allow network administrators to analyze network traffic, they can also be used to steal information being transmitted on a network. No sniffer programs had been authorized to be installed on the network. A look at the logs showed that traffic had successfully been captured from the network. This in itself was potential evidence of a crime, as it may be the indication of what amounts to an illegal wiretap. Among the laws possibly violated were the following:

- Wiretap Act, 18 U.S.C. 2511, which prohibits any person from intentionally intercepting wire communications or intentionally using or disclosing information, obtained from illegal wiretaps[5].

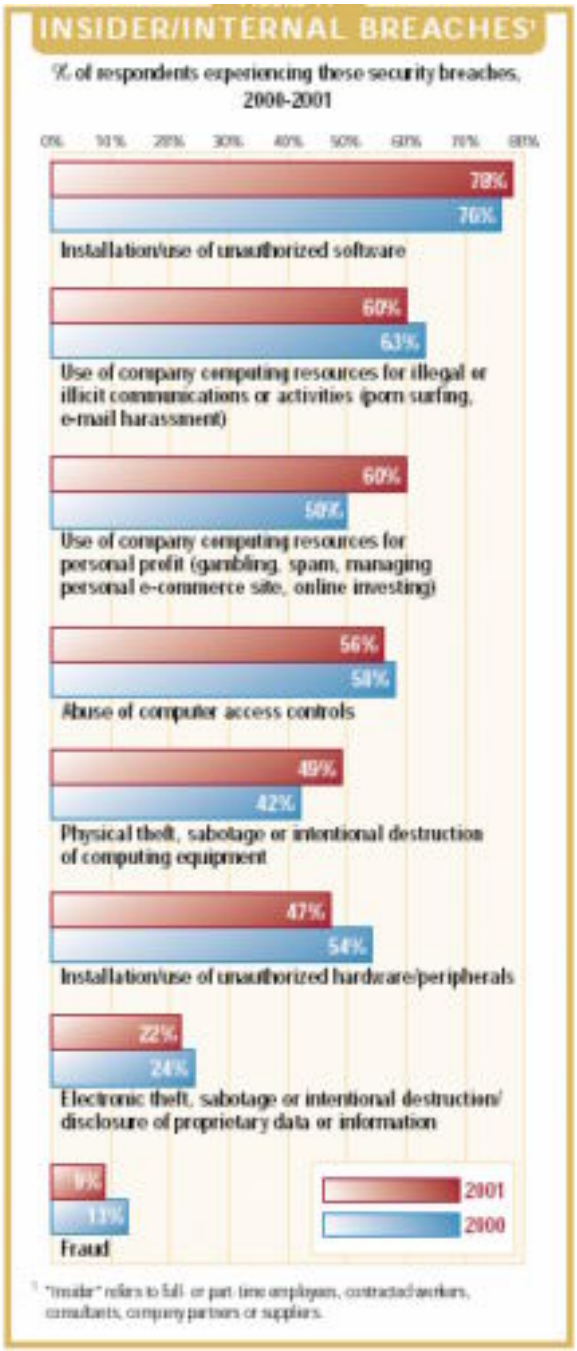
- Unlawful Access to Stored Communications, 18 U.S.C. 2701, which prohibits exceeding access authorization.[6]

It was at this point that the company's legal department was notified. They in turn notified a Law Enforcement Agency, who initiated their own investigation. In the ensuing time period of 8 months, it was discovered that an employee had indeed used Linux to run a sniffer program. The employee had been upset that a romantic relationship with another employee had ended. This person had used the sniffer program to get access to the other persons e-mail. The offending employee was eventually terminated; however, legal charges were not filed against them. The company was keen to avoid any public exposure on the security lapse, and preferred to handle the matter as an internal incident.

After action

In the aftermath of the incident, the company reviewed its legal policy. It as determined that the company could have handled things better. For starters, they did not have an effective incident response mechanism in place. They were victims of what Moira West-Brown calls the "The trial-by-fire approach". She states: "Experience shows that most organizations don't think about how to respond to a computer security incident until after they have experienced a significant one"![7] There was no mechanism in place to handle the steps necessary for incidence response. As a consequence, the initial investigation was made up as the situation unfolded. The problem lay with the fact that the company did not consider itself vulnerable to an insider attack. The argument had been given that since each employee had a background investigation, they were trustworthy. Add to this,

the fact that all sensitive systems were not connected to the Internet. The company thought that they were safe. This company unfortunately is not alone in its view. The insider threat has haunted the computer industry like never before. The recent conviction of Robert Hansen, the FBI agent turned Russian spy, is perhaps the most glaring example of the cost an insider can inflict on an organization. Consider the following statistics from the Information Security Annual Industry Survey (October 2001), a survey of 2,545 Security Practitioners: [8]



(graphic 7a)

- More than three out of four (78%) of the surveyed companies reported the installation of unauthorized software.
- More than half (60%) of companies surveyed experienced "insiders" using company computing resources for improper or illegal communications or activities.
- More than half (56%) of the surveyed companies experienced "insider" abuse of computer access controls.
- More than one in five (22%) companies surveyed experienced "insiders" engaging in electronic theft, sabotage, or intentional destruction/disclosure of proprietary information or data in 2001.
- Almost half (47%) of all respondents experienced users installing unauthorized hardware peripherals
- Almost half (49%) of all respondents experienced physical theft, sabotage or deliberate destruction of computer equipment by insiders.
- The accompanying graphic (graphic 7a) shows insider trends from 2000 and 2001.

There is a financial cost to this as well. The cost of insider abuse of net access is rising. Among the 98 respondents with quantified losses in 2001, more than $35 million was lost due to unauthorized net access costing a company an average of $357,160 per incident. Add to this the fact that more than three-fourths (76%) of network security managers said disgruntled or dishonest employees are a likely source of an attack[9]. The company itself estimated costs due to the investigation at $50,000 dollars (overtime paid to staff, etc). There were also the costs incurred to the Federal Government in its own investigation.

Lessons learned:

Often, it is mistakes that are the best teachers. Valuable lessons were learned by the company. The first realization the company made was there was a need to amend its security policy to include an incidence response capability. A team would be set up to respond to incidents. In conjunction with the company's legal department, criteria would be set forth on when it would be necessary to involve law enforcement. Funds would be made available to instruct the incident response team members (e.g. forensic training, etc). Incident handling Policy would also need to be amended. It was felt that the arguments among the staff hindered the investigation. If, for example, they would have simply asked users not to install the software again, as one staff member suggested, the company could have potentially been liable for any harassment the perpetrator would have carried out, to say nothing of protecting the privacy of employees. It was felt that if a general policy of steps taken to initiate an investigation were spelled out clearly in policy, and endorsed by management, arguments could be avoided.

Ways were also examined to tighten the security of the network. The following steps were adopted:

- Use of a BIOS password on all systems. Prior to the incident, the machines were set to boot from the CD-ROM as the default. This let any unauthorized user have access to the system. A user was able to simply boot an installation disk. This allowed them to circumvent network security, partition the hard drive, and then install the rogue operating system. The default would now be set to boot from the hard drive. The BIOS would be password protected, with only the Admin and the ISSO having access to the password. This would make it harder for another person to set the default boot sequence of a machine, and hopefully prevent a reoccurrence of the incident.

- Use of a file integrity checker. According to SANS a file checker "computes a checksum for every guarded file and stores this. At a later time you can compute a checksum again and test the current value against the stored value to determine if the file has been modified"[10]. Basically it allows you to see which files have been changed. It was felt that this step would help to pinpoint times that files were accessed, and therefore give a more accurate timestamp to any unauthorized activity. Previous to this, the only way to time stamp activity was through the conventional log. This would not tell which files a user accessed. It is possible to engage more audit features in Windows (for

example, by auditing file and object access), but it was felt that the volume of information would become unworkable. Add to this the fact that the audit wouldn't show which files had been changed, only which had been accessed. Further investigation would be necessary to determine if the file were changed (for example, a Trojan inserted). The specific package is called *Tripwire*.[11] It is intended to aid in host-based intrusion detection by checking for added, deleted, or modified files.

- A mechanism to check for sniffers. A software package called CPM (check Promiscuous Mode, for SUN OS) was utilized. The package is free software available from Carnegie Mellon. According to its homepage, the software "Checks a system for any network interfaces in promiscuous mode; this may indicate that an attacker has broken in and started a packet snooping program"[12]. This would help determine if a sniffer were being run on the network, and hopefully avoid a replay of the incident mentioned in this paper.

Clearly the addition of the above packages can help mitigate risk from insider attack, however, there is no such thing as a solution in a box. In other words, software alone would not totally eliminate risk. A true defense in depth posture was going to have to be developed. In addition to the software packets, personnel would also have to receive training. Users were all retrained on the do's and don't of system policy. In addition, security personnel were trained on potential threats, and possible indicators of violations. The staff was instructed in many of the known ways that malicious users might use to circumvent BIOS password. These include the existence of "backdoor" passwords, put into place by the manufacturers[13] (which has been mitigated by changing or eliminating the manufacturers passwords where possible), resetting jumpers or removing the CMOS battery (which has been mitigated through the use of tamper seals), and the existence of BIOS cracking utilities. Among the BIOS cracking utilities discussed were *KillCMOS*[14], which erases the CMOS settings. (KillCMOS will be detected by most virus detection software). Another utility is *RemPass*[15], a program that can find and show a BIOS password, remove a BIOS password, and save and restore BIOS settings. The software can be run in Windows or DOS mode. By helping to identify some of the risks, it was felt that the staff would be better able to identify violations, should they occur.

Conclusion

It was felt that in many ways the company was lucky. If the perpetrator had covered their tracks better (for example, not using the sniffer for a month, or deleting the sniffer logs), an investigation may not have been able to proceed. The perpetrator could also have made things more difficult for the investigators. It would have also been possible for the perpetrator to simply steal several other user's logins and passwords in order to obfuscate their identity. The perpetrator could also have ensured that the single user mode on the Linux partition was password protected. This would have expended more time from the inspectors, and therefore more cost. There were also countless other ways the perpetrator could have accomplished their goal (which was simply to read another's user's e-mail). This could have been done by keyboard capture devices, keyboard capture software, or hacking into the Mail Server directly. Discussions in these

vulnerabilities changed the paradigm the security staff had been operating in. They were isolated from the Internet, had users subjected to background checks, and had security policies in place. They thought that they were safe from attack. They were not. Clearly vulnerabilities will always exist. But the first step in dealing with vulnerabilities is often admitting the fact that you are vulnerable.

References

[1] Gaudin, Sharon; James Savage of the U.S. Secret Service quoted in an article for *Network Fusion News* http://www.nwfusion.com/news/2002/130577_03-04-2002.html

[2] Neumann, Peter G., SRI Computer Science Lab *The Challenges of Insider Misuse,* http://www.csl.sri.com/users/neumann/pgn-misuse.html

[3] Symantec Norton Ghost Homepage http://enterprisesecurity.symantec.com/products/products.cfm?productID=3

[4] Patition Magic Homepage http://www.powerquest.com/partitionmagic/

[5] 18 U.S.C. 2511.Interception and Disclosure of Wire, Oral, or Electronic Communications Prohibited http://www.usdoj.gov/criminal/cybercrime/usc2511.htm

[6] 18 U.S.C. 2701. Unlawful Access to Stored Communications http://www.usdoj.gov/criminal/cybercrime/usc2701.htm

[7] West-Brown, Moira; *Avoiding the Trial-by-Fire Approach to Security Incidents* http://interactive.sei.cmu.edu/news@sei/columns/security_matters/1999/mar/security_matters.htm

[8] Briney, Andy; *2001 Industry Survey*, infosecurity Magazine, Oct,2001 http://www.infosecuritymag.com/articles/october01/images/survey.pdf

[9]Briney, Andy; *2001 Industry Survey*, infosecurity Magazine, Oct,2001 http://www.infosecuritymag.com/articles/october01/images/survey.pdf

[10] http://www.sans.org/newlook/resources/IDFAQ/integrity_checker.htm

[11] Tripwire Homepage: http://www.tripwire.org

[12] Download site for CPM and many other security tools; http://www.mycert.mimos.my/resource/fic.htm

[13] LabMice.net *How to Bypass BIOS passwords* http://www.labmice.net/articles/BIOS_hack.htm

[14] Download site for KillCMOS http://freepctech.com/pc/002/files001.shtml#KillCMOS

[15] RemPass Home Pager; http://natan.zejn.si/rempass.html