Analysis of unknown binary, evaluation of tools SecReport and Delta and legal considerations of security incident scenario

GIAC Certified Forensic Analyst (GCFA)
Practical Assignment
Version 1.2 (December 30, 2002)

Alexander Kotkov
Submitted March 12, 2003

ABSTRACT

This paper consists of tree parts. In first part, unknown binary from compromised system was analyzed. Forensic techniques, learned at SANS training were used during this research. Second part analyses tool of my choice and evaluates its suitability for forensic purposes. I tested reliability, functionality and effectiveness of the tool. I evaluated impact created by tool on environment and status of tested system. Last part of paper discusses legal side of presented situation of incident response. Related to this situation laws are briefly discussed.

Table of Contents

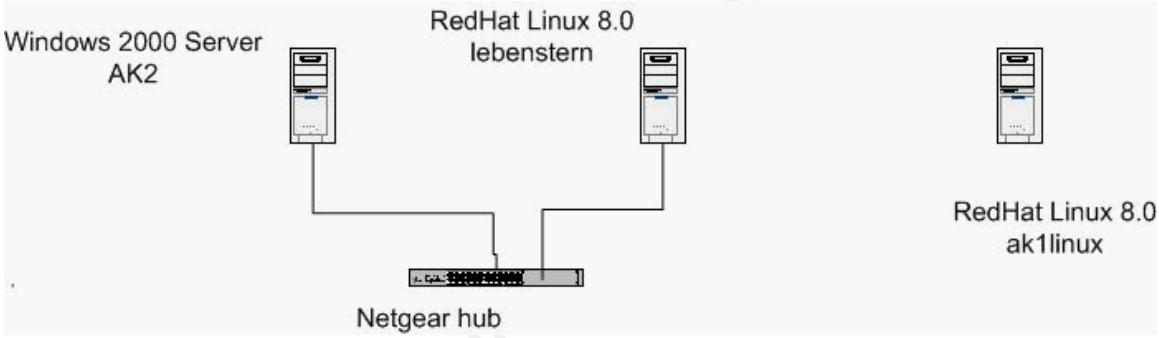
Part 1 – Analyze an Unknown Binary

Introduction

In this part of practical assignment it is necessary to analyze unknown binary, seized from compromised system. There is no known facts about functionality, origin and purpose of this binary, so it is my task as forensic investigator to find it out.

Description of test environment

For conducting tests I used setup environment configured as pictured below.



Systems were configured as follows:

- AK2: Windows 2000 Server with SP3 applied; IE 6.0;
- lebenstern – RedHat Linux 8.0
- ak1linux – RedHat Linux 8.0

All systems are:

- non-branded PC,
- Intel Pentium-III CPU 450 MHz;
- RAM: 384 MB – AK2; 256 MB – lebenstern and ak1linux

All systems were installed on "sanitized" hard disks. In order to "sanitize" (wipe out all residual information) I used floppy disk based Linux OS hal91 (created by Øyvind Kolås and located at: http://jspiro.tripod.com/linux/hal91.htm[1]).
I booted from this floppy disk and issued command:

```
dd if=/dev/zero of=/dev/hda
```

I used Windows 2000 system AK2 in order to connect to Internet for short time, using dial-up modem; download required binary files from SANS web site. After this Internet connection and modem were removed.

All systems have "private" IP addresses in class C subnet 192.168.1.0.
Computers AK1 and lebenstern are connected through 100Mb Ethernet connection (used Netgear hub).
Computer ak1linux was standalone system without any network connectivity – it was used for dynamic tests of unknown binary and as precaution was isolated from other systems and Internet.

This installation was used for conducting tests of Part 1.

Practical steps

1. I downloaded zip file binary_v1.2.zip from SANS web site and burned it on CD-R disk – in order to have unmodified copy of file on read-only media. Zip file from CD-R disk was copied to hard disk partition on forensic analysis system with RedHat Linux 8.0. Linux OS was selected as platform for analysis because it has extensive set of tools, well-suited for forensic investigation.

2. I run command:

```
zipinfo -v binary_v1.2.zip
```

Complete results of running this command are very verbose and due to space considerations were moved to Appendix 1.1.
The command outputs verbose information on contents of zip file binary_v1.2.zip
Few remarkable notices from results of this command:

- File system or operating system of origin: MS-DOS, OS/2 or NT FAT
- File security status: not encrypted
- File last modified:
  - File atd.md5: August 22, 2002, 14:58:08
  - File atd: August 22, 2002, 14:57:54
- Apparent file type:
  - File atd.md5: text
  - File atd: binary

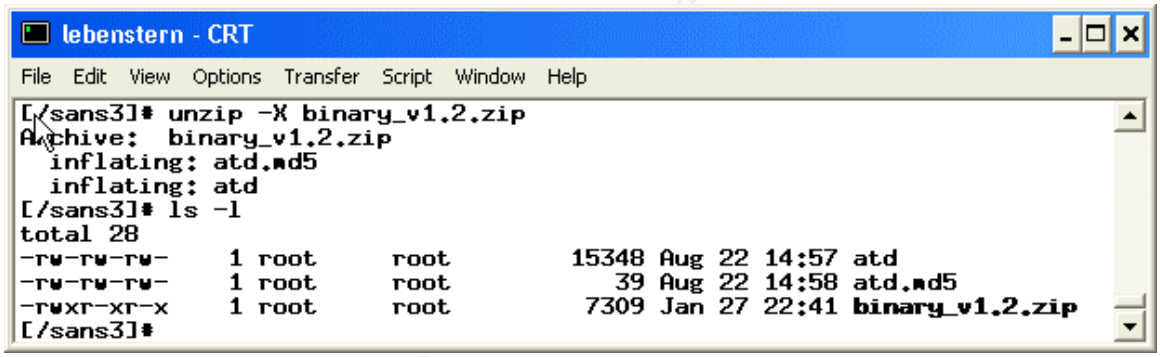
File atd has timestamp: 8/22/2002, 14:57:54; file atd.md5 time stamped 8/22/2002, 14:58:08 – time difference between file and its md5 hash file – 14 seconds. That fact, most likely can be explained that file was transferred by some means (like FTP, netcat etc) with loss of original timestamp of atd binary.

I conducted small test in order to prove this theory: created new text file on Linux system, noted MAC times of this file and after this I transferred file by FTP to Windows 2000 FTP server. Resulting file on Windows 2000 system had MAC times that were different from MAC times of original file on Linux system, but in fact were times of FTP transfer.
As soon as atd file was transferred from compromised system, person that handled incident run md5sum program in order to obtain md5 hash of original file. So, timestamp of suspicious binary unfortunately was changed and this makes much more difficult process of pinpointing time interval of initial system compromise.

3. Command:

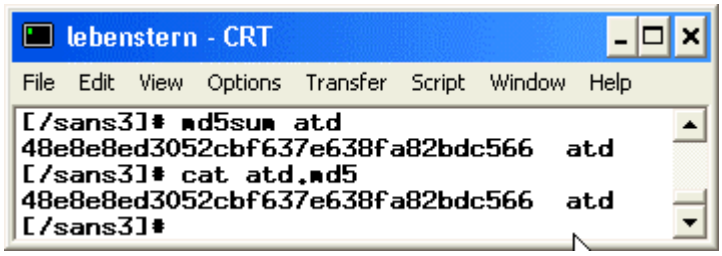
```
unzip -X binary_v1.2.zip
```

was executed in order to extract files from zip file; parameter –X forces unzipping program to restore UID/GID info for unpacked files.



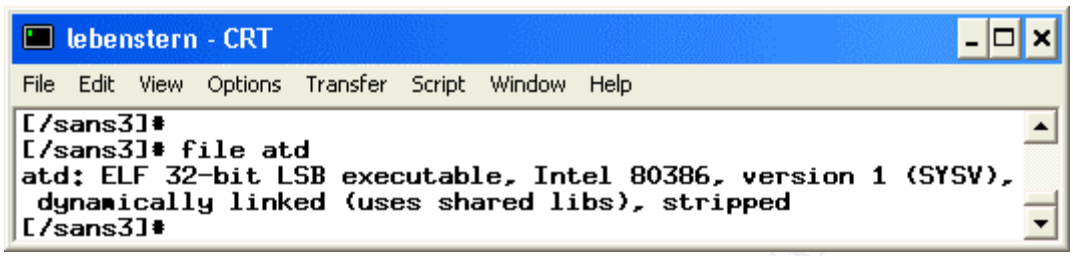
As can be seen from results of running ls –l command immediately after unpacking files from archive, both extracted files – atd and atd.md5 have file attributes –rw-rw-rw-. In Linux context that means that files have read/write access for root, group of owner and group "others" (everything that is not a root or member of file owner's group). What is very special about these results – in Unix/Linux world that means that binary files cannot be executed.

4. I executed command md5sum against file atd and printed on console text file atd.md5:



Results of these commands show that file atd.md5 contains valid md5 hash of file atd – that means, that file atd was not modified since md5 hash was created for it. That is very important and proves that we deal with original binary, seized from compromised system.

5. Next, I run command `file` in order to find out type of atd file:



Output of this command provides us with following information:

- File atd is 32-bit executable in ELF format (that is one of formats for executable files on Linux platform);
- File was compiled for execution on Intel 32-bit CPU 386 or later;
- File is dynamically linked (uses shared libraries) – that means, that for execution of some standard functions, this binary would rely on functionality of shared libraries. In other words, this is not self-contained executable file (It would be in case of statically linked executable). Generally, dynamic linking creates very compact size of binary;
- Executable was stripped – that means that after compilation, command strip atd was run – that removes all symbols from object code. It also reduces size of binary.

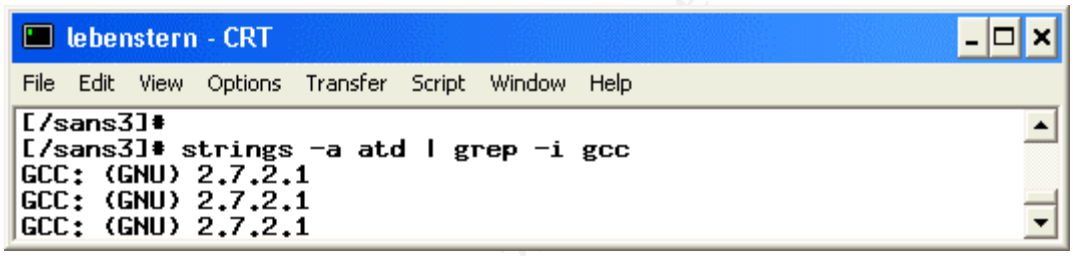
By this point, few conclusions can be done:

- We have original, unmodified binary atd (md5 hash confirms that);
- Binary was compiled in order to run on Linux OS and Intel 386 or later CPU;
- UID/GID information of file was lost or modified – binary file in order to be executed need to have "x" (eXecute) flag in its UID/GID – most likely it happened during initial incident response when file was transferred to DOS/Windows system (see results of zipinfo command, Appendix 1.1);
- Binary file atd most likely has modified timestamp of last modification – results of zipinfo command show that it was modified only 14 seconds earlier than its md5 file. It is not likely that attacker leaves around telltale evidence like md5 files for his/her binaries – most likely person that initially handled incident changed "Modified" timestamp by running some command against binary file. In 14 seconds after running this command (and modification of timestamp) md5sum command was run by incident handler – that explains 14 seconds difference of timestamp.

6. Next, I run command `strings -a -n 3`

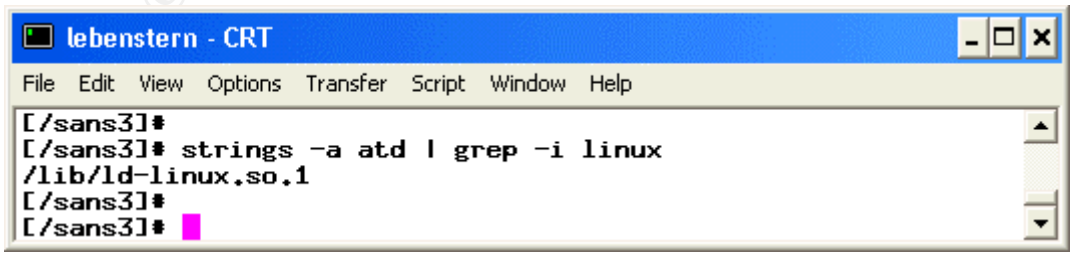
This commands searches for all NUL-terminated sequences of printable (i.e. not control or escaped) characters). Parameter –a means search through all file, not just data part. I supplied parameter –n 3 for finding all strings with length 3 or more characters (default for strings command is search for 4 or more characters long strings). In fact, I tried combinations of searching for various string lengths – setting this value to 3 yielded some valuable hints (like type of encryption – XOR – it will be described in more detail in section Compilation of Loki2). `Strings` command create very verbose, multiline output.

Raw output of strings command contained a lot of "noise" (meaningless combinations of characters that formally met filtering criteria of strings command). I filtered out those "noise" strings; also some strings were repeated 8 times in a row – I removed duplicates as well. This filtered output of strings command still has a lot of entries – I moved it to Appendix 1.2 for reference. Result of running strings command indicates that binary was compiled using GNU version of C compiler, namely gcc (GNU) 2.7.2.1. I used `grep` filter in order to highlight just relevant lines.

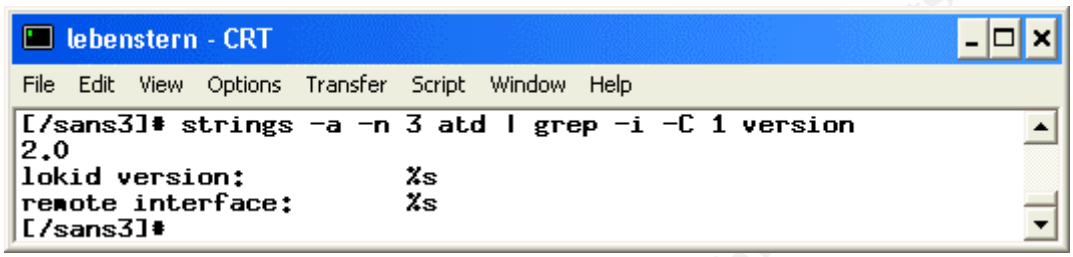


Quick search on Internet provided release date for gcc compiler version 2.7.2.1 – June 29, 1996[2]
Search on Google search engine for release dates of RedHat Linux gives idea that it could be version 3.0.4 (Rembrandt) – released July 30, 1996[3], or later version of Red Hat linux.
Additional search for RedHat Linux version released with gcc version 2.7.2.1 brought reference to RedHat 4.2 (Biltmore), released May 19, 1997[4]

Further, output of strings command (see Appendix 1.2 for full output) gives hint that platform for which binary was compiled – some version of Linux, as it contains line that looks like reference to Linux shared library:

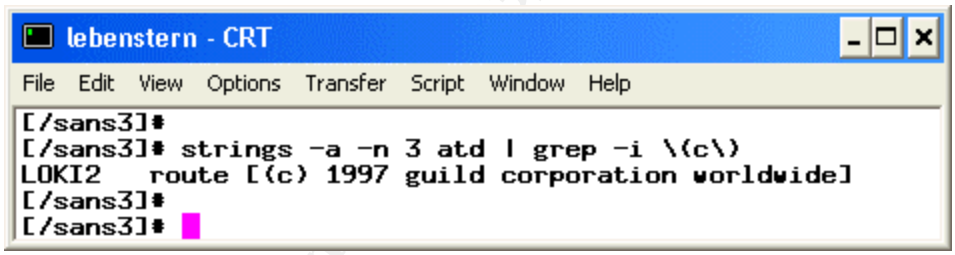


Results of command readelf –a atd also explicitly show references to Linux sahred libraries. Output of redelf command was placed in Appendix 1.3 for space considerations.
Review of strings program output (see Appendix 1.2) reveals name and version number of program: lokid 2.0. I used `grep` filter in order to highlight just relevant lines. Parameter "–i" of `grep` program makes search case-insensitive. Parameter "-C 1" means: print in context – in this case, starting one line before of line, matching filter criteria and ending one line after it.



Further scrutinizing of strings output (or use of grep filter) finally hit what can be considered goldmine in this situation: Copyright information. Specially constructed grep filter with parameters:

`-I` – for case-insensitivity;
`\(c\)` – regular expression for search of (c) – Copyright symbol.
Symbols (and) are prefixed by backslash ("escaped") because they have special meaning in regular expressions.



This output gives us:

- name of program: LOKI2
- copyright information: route [(c) 1997 guild corporation worldwide]

Search on Google search engine with keywords "loki2" "guild" lead to article "LOKI2 (The Implementation) in online Magazine "Phrack"[5].

This article (and pre-dating it article "Project Loki" in the same magazine Phrack[6] describe concept and proof of concept implementation of concealing network traffic of virtually any nature within ICMP traffic.
Normally, ICMP traffic is used for network diagnostics and troubleshooting (commands such as ping and traceroute use it). Due to this functionality ICMP traffic very often is enabled to pass firewalls, thus making it convenient transport for establishing covert channels of network traffic. In fact, this concept exploits

the fact that standard ICMP packet has data area (default size: 56 bytes) that is very seldom used and almost never analyzed by firewalls.
In fact, even RFC 792[7] that serves as standard for ICMP protocol does not specify contents of data field for ICMP_ECHO and ICMP_ECHOREPLY messages. It merely mentions that: "The data received in the echo message must be returned in the echo reply message".

Loki program uses as transport ICMP packets of type 0x0 – ICMP_ECHOREPLY (reply) and 0x8 - ICMP_ECHO (query). Program loki2 "tunnels" any other network protocol inside usually unused data field of ICMP_ECHO and ICMP_ECHOREPLY packets, creating covert channel for information exchange. Loki2 can be configured to transmit "hidden" data in plaintext (this is not safe option, as traffic can be intercepted by IDS or network sniffer) or payload can be encrypted – with simple logical XOR operation or more elaborate encryption techniques such as Blowfish and DH.
This kind of covert network traffic is quite difficult to detect. Reliable signature of this kind of exploit – unusually high level of ICMP traffic from single IP address. Network sniffer and well-tuned network IDS can be helpful in detecting this backdoor.
Timeframe of Phrack publication about LOKI2 (Published: September 01, 1997) correlates well with earlier findings about approximate timeframe of atd compilation.

Some notes on covert channels

Covert channels are quite an interesting area for computer forensic investigator. I would define it "in layman terms" as "network steganography" or "art and techniques to hide not allowed communications inside legitimate communications".
U.S. Department of Defense in document "Trusted Computer System Evaluation Criteria December" provides following definition:
"A covert channel is any communication channel that can be exploited by a process to transfer information in a manner that violates the system's security policy[8]".
Loki tool is one of relatively "old", well-known tools that exploit ICMP protocol in order to create covert channel. ICMP is not the only protocol that is being used for creation of covert information channels. There are tools that create covert channels in legitimate DNS[9], Telnet, HTTP[10] traffic.
TCP protocol also was used for creation of covert channels. Article by Craig H. Rowland "Covert channels in the TCP/IP protocol suite[11]" (http://www.firstmonday.dk/issues/issue2_5/rowland/) provides good analysis of this topic.
IP network protocol also has its share of applications that create covert channels. Article "A brief programming tutorial in C for raw sockets "[12] by Mixter (

http://mixter.void.ru/rawip.txt) provides mix of theory and samples of code that utilize IP network protocol for creation of covert channels.

Compilation of Loki2

Article in Phrack online magazine (Phrack, Vol.7, Issue 51, article 06) contains full source code of LOKI2 in C programming language. For proper functionality source code need to be extracted by special program `extract` (also provided as C language source code). I compiled `extract` program using command:

```
gcc -o extract extract.c
```

```
lebenstern - CRT
File Edit View Options Transfer Script Window Help
[/sans]# gcc -o extract extract.c
[/sans]# ls -la extract*
-rwxr-xr-x    1 root     root        13009 Jan 25 20:05 extract
-rw-rw-rw-    1 root     root         1457 Jan  3 11:32 extract.c
-rwxrwxrwx    1 root     root        62917 Jan  3 11:41 extractloki2
-rw-r--r--    1 root     root         2524 Jan  3 11:28 extract.txt
[/sans]#
```

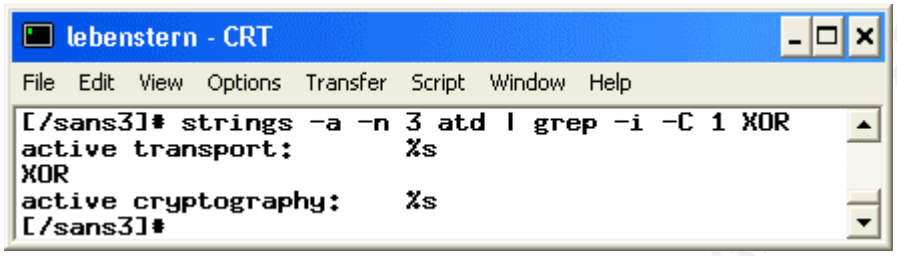
Following instructions of Phrack article, I extracted source code (in C programming language) of Loki2 program from text of article:

```
lebenstern - CRT
File Edit View Options Transfer Script Window Help
[/loki]# ./extract Phrack51-06
- Extracting L2/Makefile
- Extracting L2/client_db.c
- Extracting L2/client_db.h
- Extracting L2/crypt.c
- Extracting L2/crypt.h
- Extracting L2/loki.c
- Extracting L2/loki.h
- Extracting L2/lokid.c
- Extracting L2/md5/Makefile
- Extracting L2/md5/global.h
- Extracting L2/md5/md5.h
- Extracting L2/md5/md5c.c
- Extracting L2/pty.c
- Extracting L2/shm.c
- Extracting L2/shm.h
- Extracting L2/surplus.c
[/loki]# ls -l
total 136
-rwxr-xr-x    1 root     root        13009 Jan 25 22:22 extract
drwx------    3 root     root         4096 Jan 29 22:01 L2
-rw-r--r--    1 root     root       111823 Jan 25 22:22 Phrack51-06
[/loki]#
```

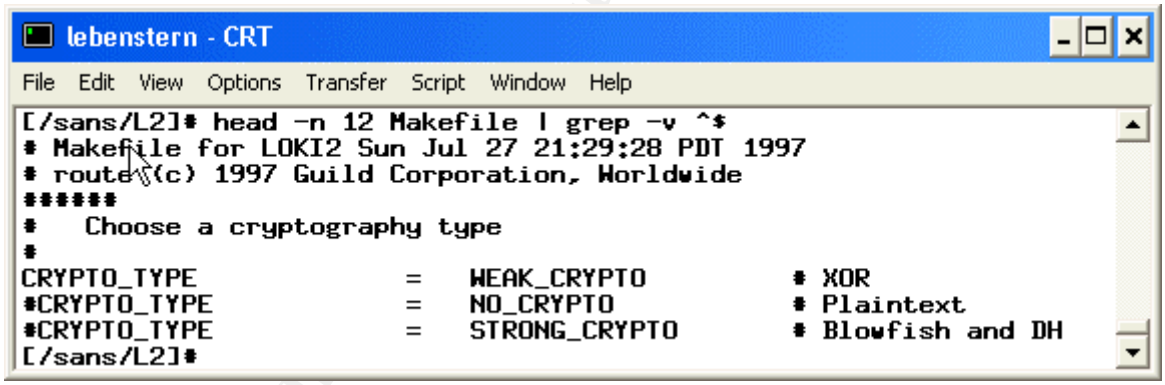
I read compilation instructions for program LOKI2, provided on Phrack web site. Instructions have few options for compiling LOKI2 program.
One of the options – encryption strength. There are 3 options available:

- STRONG_CRYPTO (DH and Blowfish)
- WEAK_CRYPTO (XOR)
- NO_CRYPTO (data is transmitted in plaintext)

Targeted review of `strings` command output (see Appendix 1) indicates that this binary was compiled with option: XOR



With this information, I left encryption type in Makefile as WEAK_CRYPTO (XOR) – in fact it was default. Below is the beginning of Makefile (that is used to provide compiler with selected options). I used grep filter with parameter "-v ^$" in order to filter out blank lines:

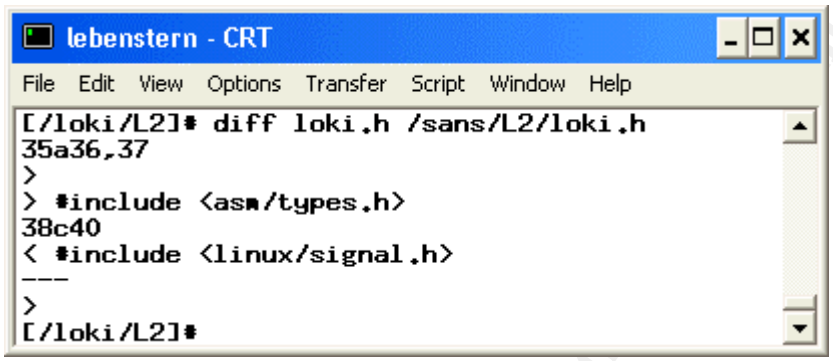


Because during string analysis I have not found clear indications of values for other adjustable parameters (Phrack article documents few others, like type of terminal and delay interval between transmissions), I decided not to change other compilation options.
In order to compile and link LOKI2 program I run command:

```
make linux
```

Program failed with number of C-compiler parsing error messages. Full output of this attempt can be found in Appendix 1.4.
Forensic investigation system has RedHat Linux 8.0 (released: September 2002) with gcc compiler version 3.2 20020903. As I mentioned earlier, binary obtained from compromised system was compiled with gcc (GNU) compiler version

2.7.2.1 that was bundled with RedHat Linux 4.2, released in May, 1997. C programming language is well standardized and provides excellent portability on source code level, but for more than 5 years between release of RedHat Linux versions 4.2 and 8.0 some minor changes in location of header types and their content could happen. After some educated guesswork and few attempts to correct the situation, I was able to compile successfully LOKI2 program on RedHat Linux 8.0 system. Modification that was required (at least the one that produced error-free compilation and linking) was editing header file "loki.h", included in source code, published in Phrack. Details of modification can be seen in output of diff command (file loki.h in current directory – unmodified, from Phrack; file loki.h in /sans/L2/ directory was modified):



Finally, running command:

```
make linux
```

resulted in successful compilation and linking of LOKI2 program – in fact, it produced two executable files – loki and lokid – as can be seen from results of running command ls –lt:

```
lebenstern - CRT
File  Edit  View  Options  Transfer  Script  Window  Help
[/sans2]# make linux
make[1]: Entering directory `/sans2'
gcc -Wall -O6 -finline-functions -funroll-all-loops -DLINUX -DWEAK_CRYPTO -DP
OPEN -DSEND_PAUSE=100 -Dx86_FAST_CHECK      -c surplus.c -o surplus.o
gcc -Wall -O6 -finline-functions -funroll-all-loops -DLINUX -DWEAK_CRYPTO -DP
OPEN -DSEND_PAUSE=100 -Dx86_FAST_CHECK      -c crypt.c -o crypt.o
gcc -Wall -O6 -finline-functions -funroll-all-loops -DLINUX -DWEAK_CRYPTO -DP
OPEN -DSEND_PAUSE=100 -Dx86_FAST_CHECK      -c loki.c -o loki.o
loki.c:347:20: warning: multi-line string literals are deprecated
gcc -Wall -O6 -finline-functions -funroll-all-loops -DLINUX -DWEAK_CRYPTO -DP
OPEN -DSEND_PAUSE=100 -Dx86_FAST_CHECK      -c client_db.c -o client_db.o
gcc -Wall -O6 -finline-functions -funroll-all-loops -DLINUX -DWEAK_CRYPTO -DP
OPEN -DSEND_PAUSE=100 -Dx86_FAST_CHECK      -c shm.c -o shm.o
gcc -Wall -O6 -finline-functions -funroll-all-loops -DLINUX -DWEAK_CRYPTO -DP
OPEN -DSEND_PAUSE=100 -Dx86_FAST_CHECK      -c pty.c -o pty.o
gcc -Wall -O6 -finline-functions -funroll-all-loops -DLINUX -DWEAK_CRYPTO -DP
OPEN -DSEND_PAUSE=100 -Dx86_FAST_CHECK      -c lokid.c -o lokid.o
gcc -Wall -O6 -finline-functions -funroll-all-loops -DLINUX -DWEAK_CRYPTO -DP
OPEN -DSEND_PAUSE=100 -Dx86_FAST_CHECK      surplus.o crypt.o  loki.c -o loki

loki.c:347:20: warning: multi-line string literals are deprecated
gcc -Wall -O6 -finline-functions -funroll-all-loops -DLINUX -DWEAK_CRYPTO -DP
OPEN -DSEND_PAUSE=100 -Dx86_FAST_CHECK      client_db.o shm.o surplus.o crypt.
o pty.o  lokid.c -o lokid
make[1]: Leaving directory `/sans2'
[/sans2]# ls -lt
total 192
-rwxr-xr-x    1 root     root        10720 Jan 29 22:21 loki
-rwxr-xr-x    1 root     root        16088 Jan 29 22:21 lokid
-rw-r--r--    1 root     root        10120 Jan 29 22:21 lokid.o
-rw-r--r--    1 root     root          579 Jan 29 22:21 pty.o
-rw-r--r--    1 root     root         2736 Jan 29 22:21 shm.o
-rw-r--r--    1 root     root         4556 Jan 29 22:21 client_db.o
-rw-r--r--    1 root     root         7720 Jan 29 22:21 loki.o
-rw-r--r--    1 root     root          846 Jan 29 22:21 crypt.o
-rw-r--r--    1 root     root         3560 Jan 29 22:21 surplus.o
-rw-r--r--    1 root     root         8018 Jan 25 10:17 surplus.c
-rw-r--r--    1 root     root          645 Jan 25 10:17 shm.h
-rw-r--r--    1 root     root         2813 Jan 25 10:17 shm.c
-rw-r--r--    1 root     root         3739 Jan 25 10:17 pty.c
drwx------    2 root     root         4096 Jan 25 10:17 md5
-rw-r--r--    1 root     root        18876 Jan 25 10:17 lokid.c
-rw-r--r--    1 root     root        14822 Jan 25 10:17 loki.h~
-rw-r--r--    1 root     root        14827 Jan 25 10:17 loki.h
-rw-r--r--    1 root     root        16720 Jan 25 10:17 loki.c
-rw-r--r--    1 root     root          470 Jan 25 10:17 crypt.h
-rw-r--r--    1 root     root         3971 Jan 25 10:17 crypt.c
-rw-r--r--    1 root     root         1750 Jan 25 10:16 client_db.h
-rw-r--r--    1 root     root         6685 Jan 25 10:16 client_db.c
-rw-r--r--    1 root     root         2651 Jan 25 10:16 Makefile
[/sans2]#
```

Comparison of files atd and lokid

File size

Binary file lokid that was created from source code for LOKI2 program has size 16088 bytes; binary from compromised system – file atd – has size 15348 bytes. Size of binaries is different, but very close. Difference in size can be explained by compilation with different version of gcc compiler; potentially files were compiled with different compilation options.

MD5 hash

Running command md5sum produced different md5 hash for files lokid and atd – it can also be explained by compilation with different versions of gcc compiler.

```
lebenstern - CRT
File Edit View Options Transfer Script Window Help
[/sans2]# md5sum lokid
93319cdb20ebae9e4a297e09e14ad67a  lokid
[/sans2]# md5sum /sans3/atd
48e8e8ed3052cbf637e638fa82bdc566  /sans3/atd
[/sans2]#
```

File. Strings

In order to indirectly compare files, I run file and strings programs against lokid binary. Output of file command was identical for files lokid and atd:

```
lebenstern - CRT
File Edit View Options Transfer Script Window Help
[/sans2]# file lokid
lokid: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
dynamically linked (uses shared libs), stripped
[/sans2]# file /sans3/atd
/sans3/atd: ELF 32-bit LSB executable, Intel 80386, version 1 (SY
SV), dynamically linked (uses shared libs), stripped
[/sans2]#
```

Running strings –a –n3 command against binary lokid produced results, almost identical to running strings –a –n 3 on atd binary. Full list of strings, obtained from lokid file can be found in Appendix 1.6. Meaningful strings that were different are those related to version of gcc compiler, which should be expected (as I mentioned earlier, I used gcc compiler provided with RedHat Linux 8.0, while file atd was compiled with older version of gcc compiler).

Strace

Result of execution command strace ./lokid can be found in Appendix 1.7. Review of results obtained from execution of same command against file atd (results are in Appendix 1.5) shows a lot of correlation between results.

Banner

Banners of binary files atd and lokid are identical, as screenshots below indicate.

```
165.181.216.41 (5) - CRT
File Edit View Options Transfer Script Window Help

[root@lebenstern sans2]# ./lokid

LOKI2   route [(c) 1997 guild corporation worldwide]
[root@lebenstern sans2]#
Ready  Telnet  2, 26  7 Rows, 58 Cols  VT100
```

```
root@ak1linux:~/sans
File  Edit  View  Terminal  Go  Help
[~/sans]# ./atd

LOKI2   route [(c) 1997 guild corporation worldwide]
[~/sans]#
```

Comparison, conducted by using techniques and tools described above can serve as indirect prove that atd and lokid are most likely essentially the same program. Minor differences between can be attributed to different versions of OS and gcc compiler.

Dynamic analysis of atd binary

Based on results of running readelf –a atd command (see Appendix 1.3 for complete printout), I realized that it will require libc.so.5 and ld-linux.so.1 shared libraries. According to article at RPMfind.net[13]:
"Older Linux systems (including all Red Hat Linux releases between 2.0 and 4.2, inclusive) were based on libc version 5. The libc package includes the libc5 libraries and other libraries based on libc5. With these libraries installed, old applications which need them will be able to run on your glibc (libc version 6) based system."
As I found earlier, binary atd most likely was compiled for RedHat Linux 4.2.
In order to provide atd binary with necessary shared libraries, I downloaded from the same rpmfind.net web page necessary RPMs for shared libraries and installed them on stand-alone PC with RedHat Linux 8.0 – I decided to use separate PC, disconnected from any network and Internet in order to avoid potential contamination of other systems by unknown binary atd.

Before executing atd binary on test system, I run few commands in order to obtain "pre-run" information.

1. Executed command:

```
ps -Af
```

This command provides information about all processes, currently running on system. Parameter `f` prints out information in "full" format. Output was long enough and I do not include it because of space considerations. Interestingly enough, I noticed that process with name atd was listed as running on the system.
First command issued:

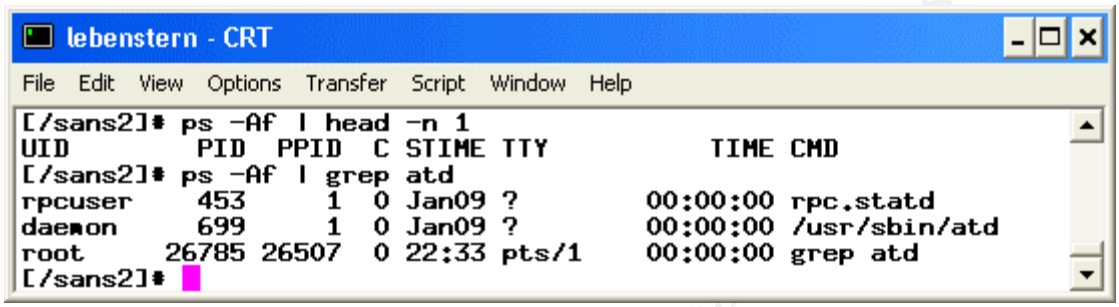
```
ps -Af | head -n 1
```

is provided for convenience – it types column headers for subsequent output.

Next command:

```
ps -Af | grep atd
```

filters out only line that contain search criteria "atd"
As can be seen, 3 processes match this search criteria; of these 3 processes process with PID = 699 is direct hit – it points to program that was started as `/usr/sbin/atd`



Quick search on Internet showed that atd is daemon for scheduling program at – it is simpler, but less flexible scheduler than cron is. It allows scheduling programs for single execution only and has very simple syntax, like[14]:

```
at 13:00 today
at 1 PM today
at noon tomorrow
at 23:00 19.02.03
```

This discovery gives another hint – person that installed atd binary:

- Has decent knowledge of Linux system – most likely he/she is not a "script kiddie";
- Wanted to conceal this process by naming (or renaming) it to name of standard Linux daemon – that leads to conclusion that it was not just harmless toy or experiment.

2. Assigning "Execute" flag to atd binary

As I noted earlier, atd file does not have x flag, so it cannot be executed (most likely, file attributes were modified during initial incident response).
I run command:

```
chmod 777 atd
```

– this command assigns all permissions (read, write, execute) to all users (root, group, other):

```
root@ak1linux:~/sans
File  Edit  View  Terminal  Go  Help
[~/sans]# chmod 777 atd
[~/sans]# ls -l atd*
-rwxrwxrwx    1 root     root        15348 Aug 22 14:57 atd
-rw-rw-rw-    1 root     root           39 Aug 22 14:58 atd.md5
[~/sans]#
```

3. I run atd program by typing command:

```
./atd
```

```
root@ak1linux:~/sans
File  Edit  View  Terminal  Go  Help
[~/sans]# ./atd

LOKI2   route [(c) 1997 guild corporation worldwide]
[~/sans]#
```

Program showed banner, that confirms initial assumption that atd binary is renamed lokid program. Banner exactly matches string that can be found in C source code of loki2 program (that produces two binaries – server – lokid and client – loki).

4. Execution of command

```
netstat -lp_atd
```

shows that atd program opened two raw network sockets – one of them listens for icmp protocol.

```
root@ak1linux:~/sans
File  Edit  View  Terminal  Go  Help
[~/sans]# netstat -lp | grep atd
tcp        0      0 *:32768                 *:*                     LISTEN      530/rpc.statd
udp        0      0 *:32768                 *:*                                 530/rpc.statd
raw        0      0 *:icmp                  *:*                     7           1036/atd
raw        0      0 *:255                   *:*                     7           1036/atd
[~/sans]#
```

5. Running command:

```
ps -aux
```

shows that PID of process for atd – 1036.

```
root@ak1linux:~/sans
File  Edit  View  Terminal  Go  Help
[~/sans]# ps -aux | grep atd
rpcuser    530  0.0  0.2  1444  724 ?        S    22:08   0:00 rpc.statd
daemon     764  0.0  0.2  1292  524 ?        S    22:08   0:00 /usr/sbin/atd
root      1036  0.0  0.1   936  348 ?        S    22:20   0:00 ./atd
root      1062  0.0  0.2  1640  656 pts/1    T    22:27   0:00 less readelf_atd.
root      1105  0.0  0.2  3204  616 pts/0    S    22:39   0:00 grep atd
[~/sans]#
```

6. Execution of command:

```
lsof -p 1036
```

shows files that are opened by process with PID 1036 (in this case, atd):

```
root@ak1linux:~/sans
File  Edit  View  Terminal  Go  Help
[~/sans]# lsof -p 1036
COMMAND  PID USER   FD   TYPE DEVICE    SIZE    NODE NAME
atd     1036 root  cwd    DIR    3,2    1024   26521 /tmp
atd     1036 root  rtd    DIR    3,2    1024       2 /
atd     1036 root  txt    REG    3,2   15348   65481 /root/sans/atd
atd     1036 root  mem    REG    3,2   25386   73542 /lib/ld-linux.so.1.9.5
atd     1036 root  mem    DEL    0,4         1015827 /SYSV000004fd
atd     1036 root  mem    REG    3,5 2176218  242881 /usr/i486-linux-libc5/lib/libc.so.5.4.44
atd     1036 root    1u   CHR  136,1               3 /dev/pts/1
atd     1036 root    2u   CHR  136,1               3 /dev/pts/1
atd     1036 root    3u   raw                   4610 00000000:0001->00000000:0000 st=07
atd     1036 root    4u   raw                   4611 00000000:00FF->00000000:0000 st=07
[~/sans]# _
```

We can see from results of running this command, that atd opens two raw connections (sockets) and uses few system libraries.

7. I killed process with PID 1036 by typing command:

```
kill -1036
```

After this I verified with command:

```
ps -aux
```

that process was terminated.

8. Running command:

strace ./atd

shows system calls that are executed by program (complete printout of ps – aux command can be found in Appendix 1.5). Interesting lines:

- 57 and 59 – create SOCK_RAW for IPPROTO_ICMP and PPROTO_RAW;
- 66 – prints program banner LOKI2.

Legal Implications

Installation and use of program atd (that was identified as renamed server part of Loki2 program – lokid) violates number of criminal laws of United States.
It qualifies as offence according to United States Code (U.S.C.) § 2701 (a)(1),(2)[15]. Person that committed this offence can be punished with:

"(1) if the offense is committed for purposes of commercial advantage, malicious destruction or damage, or private commercial gain--
(A) a fine under this title or imprisonment for not more than one year, or both, in the case of a first offense under this subparagraph; and
(B) a fine under this title or imprisonment for not more than two years, or both, for any subsequent offense under this subparagraph; and
(2) a fine under this title or imprisonment for not more than six months, or both, in any other case".[15]

Another applicable law in this situation – U.S.C. § 1030 (a)[16] qualifies this activity as fraud. Same section of law specifies more strict punishment if compromised computer belongs to any department of US government, financial institution, card issuer (generally referred as "protected computer"). Depending on severity of crime and additional circumstances (first or subsequent violation of this type; estimated cost of damage; if computer is "protected" or not) – punishment can be fine or various terms of imprisonment – from 5 to 20 years.

In case this program would be installed in my company, it qualifies as violation of Corporate IT Security policy that explicitly prohibits:

- installation of any remote access software or equipment, unless it is approved by IT management and IT Security department;
- installation of any software, unlicensed to Company;

Violation of Corporate IT Security policy similar to this serves as ground for immediate and unconditional termination of employee.
If this violation would resulted in substantial material loss (threshold value of loss is determined by Company's management), criminal investigation can be initiated. In this case employee, that installed tool similar to loki would face imprisonment and/or fine according to laws described above.

Interview questions

1. You seem to be very charismatic and intelligent person. Usually people like you have some hobbies or interests. What are you hobbies ? Are they technical in nature ?

Comment: I asked this question just to get general idea about interests of person. In case he/she indicates that hobbies are technical in nature, it is good start to continue with more detailed technical questions.

2. Let's consider the situation: you work as system administrator for this company and you live in suburban (about 2 hours of travel time from company office). Sometimes "things happen" and something does not work on your corporate network – at 2:15 am. Situation repeated few times. What would you do in order to provide prompt troubleshooting but not go every other night to your office ?

Comment: This question was asked in order to "probe" person's "mental readiness" to bypass security policy as trade-off for convinence.

3. [Question in case suspect works in company, where system was compromised]:
   All workstations and servers in your company are Windows 2000 or Windows XP. Why this small PC runs some version of Linux ?

Comment: It is quite direct question, that can bring interesting answers. Non verbal communications of suspect can be very indicative after this question asked.

4. A lot of software and excellent tools exist for Linux. Some of them easy to install, but some of them you need to compile or even tweak a little to get them working. Have you ever do something like this ?

Comment: Another probe of technical skills [that are required to install Loki].

5. This company has firewalls installed. How would you test their reliability ?

Comment: Question targets to find out level of "creativity" of person, that could result in Loki installation.

6. Have you practically tested reliability of your firewalls ?

Comment: That is direct follow-up to Question 5.

Conclusion

Results of research allow me to make following conclusions:

- Binary atd that was found on compromised system – renamed server of remote access tool Loki2

- Date when atd was installed on system is unknown; also unknown if it was executed on the system and when it was executed last time – this information cannot be restored, as timestamp of file was modified
- User/group ID that was used to install this tool is unknown, as UID/GID info was lost during initial incident response
- Tool most likely was installed by person who has decent knowledge of Linux;
- Tool was renamed as atd in order to conceal it – atd is name of legitimate Linux daemon (scheduler);
- Tool was used (or intended) to be used as tool for establishing covert network channel, capable to bypass most of firewalls.

This research shows that not following basic rules during initial incident response caused lost of valuable information about fact of execution of tool. That can (and most likely will) affect both investigation and possible trial, unless there is other evidence that can be combined with data obtained during my research and serve as basis for more complete investigation.

Part 2 – Option 2 – Perform Forensic Tool Validation

Introduction

I will evaluate as forensic tools small suite of two programs: SecReport and Delta.
Programs were written by me with following purposes:

- SecReport – to collect standard, structured set of security-related information from Windows systems;
- Delta – to compare any two reports, created by program SecReport and highlight only differences between them.

In tool SecReport I tried to implement some of recommended steps for collecting information during initial incident response.[17],[18].

Usually investigator uses number of command-line tools in order to generate reports about variety of important (or potentially important) details of cyber-crime scene. In most cases, each command line tool generates output in proprietary format that is understandable to specialists, but not always that clear to non-technical people. Due to nature of forensic investigators' activity, reports, obtained during initial response often need to be presented to non-technical audience, such as law enforcement officers, attorneys, executives etc.
I attempted to create tool that collects same information, that can be obtained by variety of tools, but presents it in more structured, intuitive and logical order, that can be relatively easy understood by non-technical audience. As design constraint, I tried to make reports concise and "readable", so certainly it is not "all-in-one" tool, but rather "many-in-one" instrument for initial collection of information for investigation of computer crime. I intentionally did not include such crucial elements, as MAC times of files, dumps of Event Log, Registry etc – just because these are huge amount of data that will make this report unreadable by non-technical people. Another consideration – I tried to use standard and "reputable" software for presentation and printing of reports. People (judges included) tend to trust tools and software they know, use and understand. Internet Explorer is one of popular applications, that is familiar to many people, so I decided to use it for displaying and printing information, collected during initial incident response. These considerations were implemented in tool SecReport.

Secondary design objective was providing means of computer-based comparison of any two reports. The only practical method for comparison of data, obtained by virtually any command line tool from two systems is line-by-line review of reports (unless some script or program exists to do so). From practice I realized that task of comparison data from different systems (or same system, at different moments) is quite common activity for security investigator. And, strangely

enough, it is not automated (generic "file comparators" are not suited well for tasks like this). Result of this idea – tool Delta that is included in evaluated suite.

These tools proved to be useful both during day-to-day security administration of medium-size company and during initial incident response – as data collection and analysis tools.
Tools can be downloaded by typing URLs:
http://members.verizon.net/~vze3vkmg/tools/getinfo.zip or
http://kotkov.tripod.com/getinfo.zip or
Current version of tools: 3.03.07.
Tools are console (command line) programs written in Visual Basic 6 programming language.
Zip file getinfo.zip (size: 807 Kbytes), mentioned above contains few auxiliary files and programs that required for functionality of tools. Here is brief description of each file:

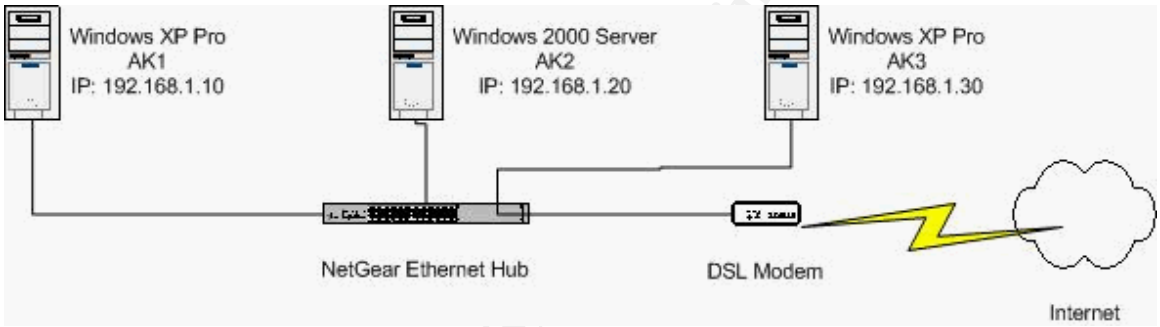
1. SecReport.exe - tool SecReport – collects information from Windows system
2. Delta.exe - tool Delta – compares 2 reports, produced by SecReport
3. securityreport.xsl - auxiliary file for SecReport tool (provides HTML-like formatting and layout for user-friendly display and printing in viewer (IE 6.0 or later)
4. deltarep.xsl - auxiliary file for Delta tool (provides HTML-like formatting and layout for user-friendly display and printing in viewer (IE 6.0 or later)
5. auditpol.exe[19] - Command line tool from MS Windows 2000 Resource Kit – used for collection of information on audit policy settings
6. fport.exe[20] - Free command line tool from Foundstone.com – used for obtaining information about mapping of open network ports to processes
7. mbsacli.exe[21] - Free command line tool Microsoft Baseline Security Analyzer (MBSA) – used for obtaining information about installed hotfixes
8. hfdll.dll - Library for MBSA – part of MBSA distribution;
9. mssecure.xml[22] - Data file with information on MS hotfixes – downloaded from Microsoft (required by MBSA)
10. ToHtml.Bat - Supplementary script for conversion of single XML report into HTML format
11. AllToHtml.Bat - Supplementary script for conversion of multiple XML reports into HTML format
12. forfiles.exe[23] - Command line tool from MS Windows 2000 Resource Kit – used for conversion of multiple XML reports into HTML format (called from AllToHtml.Bat script)
13. msxsl.exe[24] Free command line tool for XSL transformation of XML file into HTML (called from ToHtml.Bat) – download from Microsoft
14. Readme.txt - Brief instructions on usage of tools

15. GetInfo.md5 - MD5 hashes for all files included in package.

Note:
Files 1-9 are required for normal functionality of tools;
Files 10-13 are optional and required only for conversion of XML reports into HTML format (for purpose of viewing/printing reports from any browser that supports HTML 4). These files are not needed if IE 6.0 or later version is used for viewing/printing reports.
Files 14, 15 supplied as documentation.

Description of test environment for testing evaluated tools

For conducting tests I used setup environment configured as pictured below.



Systems were configured as follows:

- AK1:
  - Windows XP Professional with SP1 applied; IE 6.0-SP1;
  - non-branded PC,
  - Intel Pentium-MMX CPU 233 MHz;
  - RAM: 256 MB;
- AK2:
  - AK2: Windows 2000 Server with SP3 applied; IE 6.0-SP1;
  - Non-branded PC;
  - Intel Pentium-III CPU 450 MHz;
  - RAM: 384 MB;
- AK3:
  - Windows XP Professional with SP1 applied; IE 6.0-SP1;
  - Dell Dimension XPS T450;
  - Intel Pentium-III CPU 450 MHz;
  - RAM: 384 MB.

All systems were installed on "sanitized" hard disks. In order to "sanitize" (wipe out all residual information) I used floppy disk based Linux OS hal91 [1] (created by Øyvind Kolås and located at: http://jspiro.tripod.com/linux/hal91.htm).

I booted from this floppy disk and issued command:

```
dd if=/dev/zero of=/dev/hda
```

All systems have “private” IP addresses in class C subnet 192.168.1.0.
All computers are connected through 100Mb Ethernet connection (used Netgear hub). Systems AK1 and AK3 are configured for Internet access (through DSL connection); system AK2 is not configured for Internet connectivity.
All systems were populated with variety of applications in order to provide enough information for conducting tests.

This installation was used for conducting tests of Part 2.

Scope of testing \f C \l 3

Tool SecReport collects wide variety of security-related information (detailed list of information, collected by tool, can be found in section: Information, collected by tool SecReport). In my testing I verified every item on report by alternative means. Where possible, I used Windows GUI or command line tools. When there is no appropriate tool, provided by OS, I used reputable tools, such as Windows Resource Kit utilities and third-party tools. Main objective of this testing was to prove validity of information, repeatability and reproducibility of results.

Tool Delta compares any two reports, produced by tool SecReport and generates report only on differences between systems. I was not able to find any other tool that does similar comparison (that is why I created the tool). I manually compared results reported by tool with actual reports from two target systems, that were compared. Objective of this testing was to verify validity of presented results. Repeatability and reproducibility in this case was not tested as manual verification was the only available alternative.

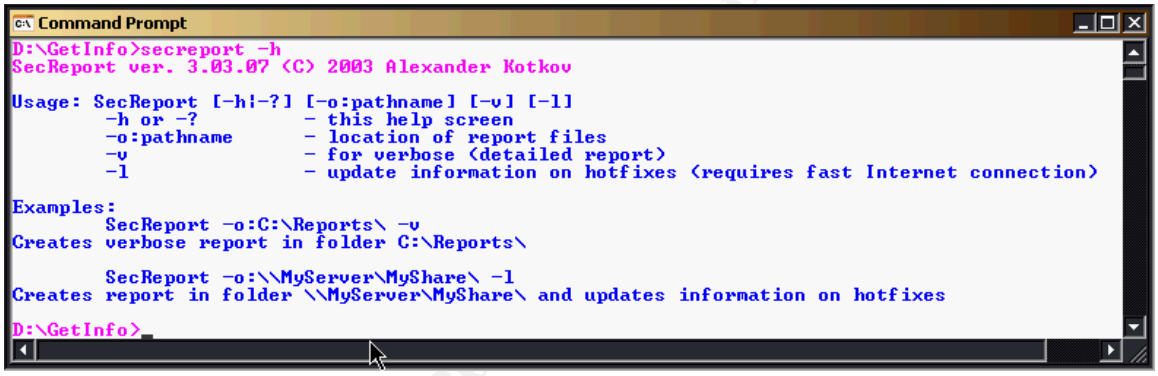
Tool SecReport

Description of tool

Supported platforms:
- Windows 2000;
- Windows XP;
- Windows 2003 (RC2).

Supported viewer:
- MS Internet Explorer 6.0 or later;

Tool accepts few command line options. Brief description of options can be obtained by typing from command prompt: secreport –h or secreport -?.



Note: screenshot above was processed with Invert Colors operation in Microsoft Paint . Original output has black background, so it would be hardly visible at printout. Similar operation was conducted on few other screenshots that required that. Displayed data was not changed.

Command line options:
-h or -? – help screen
-o:pathname – location of report files (local drive; mapped network share or UNC path);
- v – verbose (detailed report);
- l – update information on hotfixes (this option requires fast Internet connection).

For data collection during incident response and forensic investigations it is recommended to specify location for report files either on removable media (floppy or zip disk) or mapped network share or UNC path – that will avoid creating any files on local hard disk partitions. By default, if reports are not redirected (by using command line option: -o), report files will be placed into root of %SystemDrive% (which is in most cases means C:\ ).

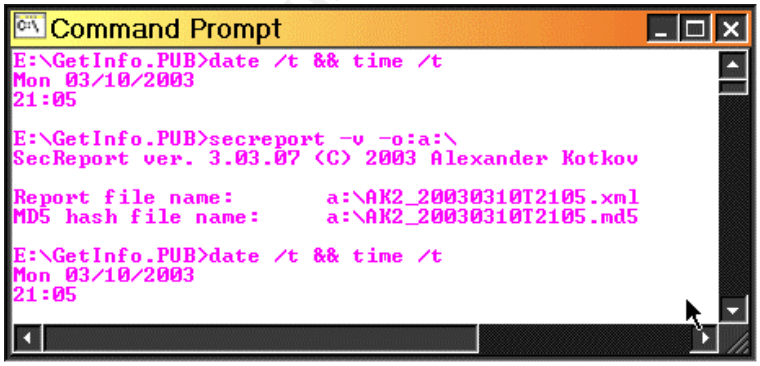
Tool creates three files:

- Report file – its filename is created by following template:
  Computername_YYYYMMDDThhmm.xml
  Where *Computername* – name of Windows system (NetBios name or host part of fully qualified DNS name);
  YYYY – 4-digit year of tool execution;
  MM – 2-digit month of tool execution (it has "leading zero" format, so for months: January - September – it will be corresponding numbers: 01 – 09;
  DD – 2-digit day of tool execution (it also has "leading zero" format, so for days 1-9 – it will look like: 01 – 09);
  hh – 2-digit hour of tool execution (in "military" time format, 0-23, with "leading zero");
  mm – 2-digit minute of tool execution (with "leading zero").

  For example, report for computer with hostname *AK1*, in case tool SecReport was executed March 10th, 2003, at 21:05, filename of report will be:
  `AK1_20030310T2105.xml`

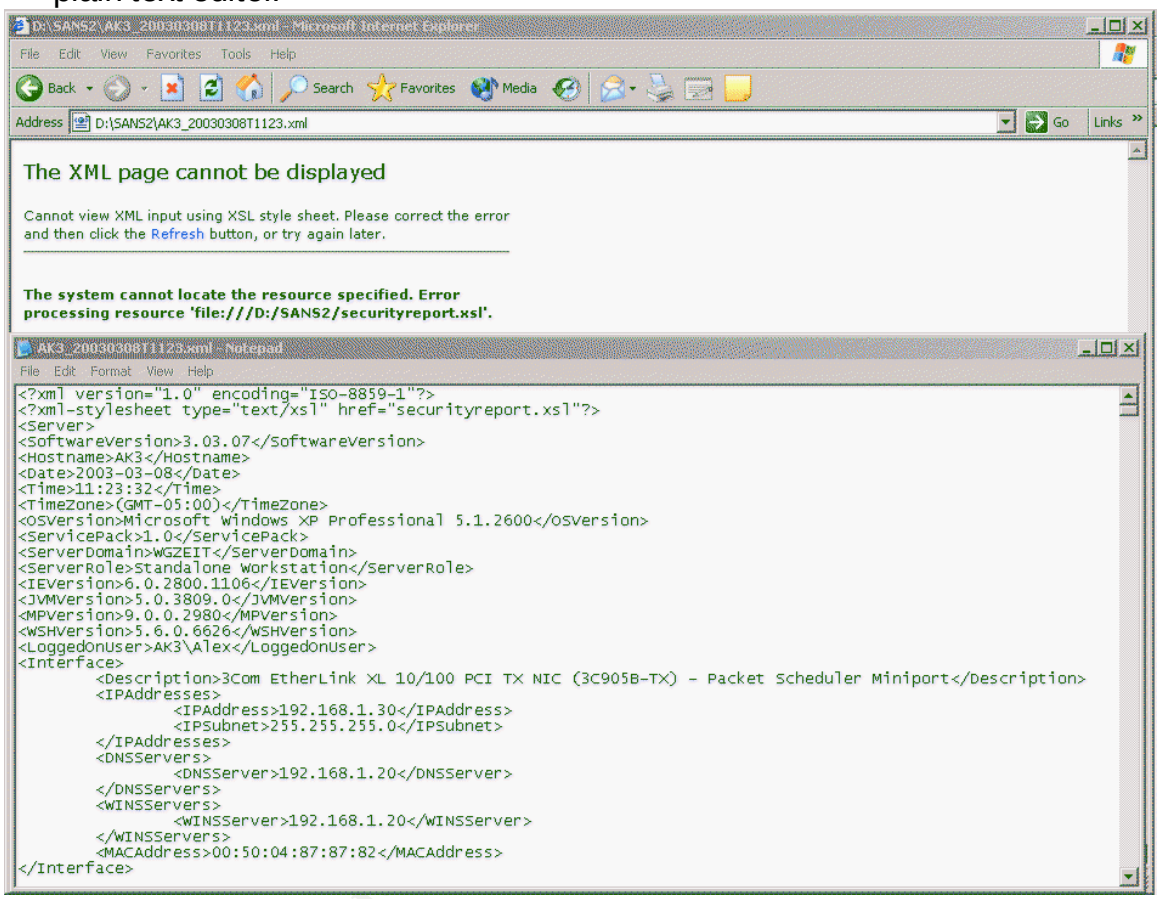
Note: Date and time format used for naming report files conforms with recommendations of ISO 8601 standard (description of this standard can be found in article written by Markus Kuhn: "A Summary of the International Standard Date and Time Notation" [25] - : http://www.cl.cam.ac.uk/~mgk25/iso-time.html).

- File with MD5 hash of report file. This file has same filename, as report file, but extension .md5. For example, md5 file for report on computer with hostname *AK1*, in case tool SecReport was executed on March 10th, 2003, at 21:05, filename with md5 hash of report file will be:
  `AK1_20030310T2105.md5`
  Creating of MD5 hash file from all data obtained during incident response and forensic investigation is a standard practice. Generating MD5 hash allows to create credible reports and serve as supporting information for maintaining "chain of custody" for reports.



- Auxiliary file *securityreport.xsl* – this is file-translator. It needs to be present in same directory where report file(s) placed. Tool will work without it, but report file will look as raw xml – which is more "machine-readable" format, rather than "human-readable". Screenshot below illustrates situation when

securityreport.xsl file is missing: browser gives an error message. If select option (in IE 6.0 browser): View – Source, raw XML report will be opened with Notepad (or other default editor). Report is still readable, but definitely it is not "user-friendly" because of XML tags (they are similar to HTML tags in syntax). Alternatively, report file can be opened and printed in raw xml format from any plain text editor.



There are three main reasons that XML was selected as format for report files:

- XML provides "self-describing" data. As can be seen from screenshot, report surrounds atomic pieces of information with tags such as <HostName> or <LoggedOnUser>. Review of raw xml report can be useful when report is challenged in court, as it eliminates "translation" part (that is provided by *securityreport.xsl* file) and represents by itself "first-generation" data.
- By modifying "translation" file *securitreport.xsl* it is possible to change look of "human-friendly" report without changing data in report itself. That can be useful for presentation of report to different audience (technical vs. non-technical etc) or for eliminating details that are not relevant for particular case.
- XML format of data is very convenient for processing by software (and presented suite of tools uses this functionality – it will be addressed in detail in section about Delta tool).

Report files need to be transferred, if necessary (it is necessary in case of incident response) to other system for storage and analysis.

When report file and "translating file" securityreprt.xsl are placed into same directory, report can be viewed by clicking on report file *Computername_YYYYMMDDThhmm.xml*. In the rest of this document, unless specifically noted, by word "Report" I will refer to file *Computername_YYYYMMDDThhmm.xml*. Clicking on report file opens it with Internet Explorer (this is Windows system default). If system configured with non-default settings and associates files with extension .xml with some other software, report can be viewed by right-clicking on report file and selecting menu option: Open With…- Internet Explorer. Systems with IE 6.0 or later will open properly formatted report. Report can be printed from IE 6 by issuing standard print command from browser: File – Print.
In order to prevent printed report from being truncated, it is recommended to set font size of browser to Smallest (IE 6 command: View – Text Size – Smallest) before printing.

Windows system, where SecReport tool executed (in further text I will refer to them as Target system) can have any version of Internet Explorer (Windows 2000 was originally shipped with IE 5.0, so in reality that is oldest possible version of IE on Windows 2000 platform; Windows XP was shipped with IE 6.0).

Information, collected by tool SecReport

Sample report (placed due to space considerations in Appendix 2.1), indicates that following information was collected from target system:

- Hostname (NetBios name or host part of full DNS name);
- Timestamps of tool execution (start and finish);
- Timezone setting of system;
- Operating system (detailed version number);
- Service pack level;
- Server domain (NT domain);
- Server role (domain controller, standalone server, workstation etc);
- IE version (detailed version number);
- Java Virtual Machine (JVM) version (detailed version number);
- Media Player version(s) – some systems can have more than one version of this software installed;
- WSH version (WSH - Windows Script Host);
- Logged on user name;
- Network configuration (NIC card(s) brand and model; IP Address(es); default gateway; IP subnet mask; DNS server(s); WINS server(s); MAC address);
- Audit policy settings. To collect audit policy settings, tool uses command line tool auditpol.exe [19] from Windows 2000 Resource Kit.
- Event log settings (size, type of overwrite, eventlog filename);

- Applications installed;
- Microsoft security hotfixes, explicitly installed on system. For collection of information about installed hotfixes, tool parses output of free Microsoft command line tool mbsacli.exe (formerly known as hfnetchk)[26].
- Services – each server described by its name, startup type, current status, service full (descriptive) name; service account;
- Basic configuration information for Internet Information Server (IIS), if it is installed:
    - names of accounts for Anonymous internet access and WAP (they have standard naming template, but often renamed for security considerations);
    - strengths of passwords for these accounts – passwords that have length 7 characters or less are are displayed; all other passwords substituted by string of 14 asterisk characters (default mode). If "verbose" mode specified (tool called with parameter: –v) then any passwords (for IIS anonymous accounts) are displayed;
    - information about location of IIS log directory;
    - version of URLScan[27], if installed;
    - information on ISAPI filters, their priority and order of their load;
    - DLL application mappings information – this kind of information allows at glance determine if system is potentially vulnerable to such exploits as CodeRed and Nimda;
- Ports (network) open – number of port, protocol (TCP or UDP), Process ID (PID) of process that owns this port, short name of process and full path to program. For collection of information about open network ports and processes that own them, tool parses output of free command line tool FPort.exe ver. 2.0, that can be downloaded from Foundstone at http://www.foundstone.com/knowledge/intrusion_detection.html [13]
- Active processes – process name; process ID (PID), process ID of parent process, thread count, handle count, command line that started process. For some system processes command line property is not applicable;
- Page file settings – location and filename(s); initial size, maximum size;
- Hardware information:
    - Brand of computer system;
    - Model;
    - Serial No. of computer;
    - Number of processors;
    - BIOS version;
    - BIOS date;
    - RAM size;
- Details for each CPU – ID, Manufacturer; Type, CPU performance, MHz, L2 cache size, clock rate of CPU's external bus;
- Logical disks – drive letter; description; filesystem; total size; free space, volume name (for mapped network drives – name of mapped share); serial No. of volume (assigned by OS);

- Physical disks: device ID, model, interface, size, number of partitions, number of bytes/sector, number of sectors/track, number of cylinders, number of heads, number of sectors, number of tracks, number of tracks/cylinder.
- Presence of installed Recovery Console;
- Date of Norton Antivirus signature[28] (in case this software is installed).;
- Name and version of software that generated report.

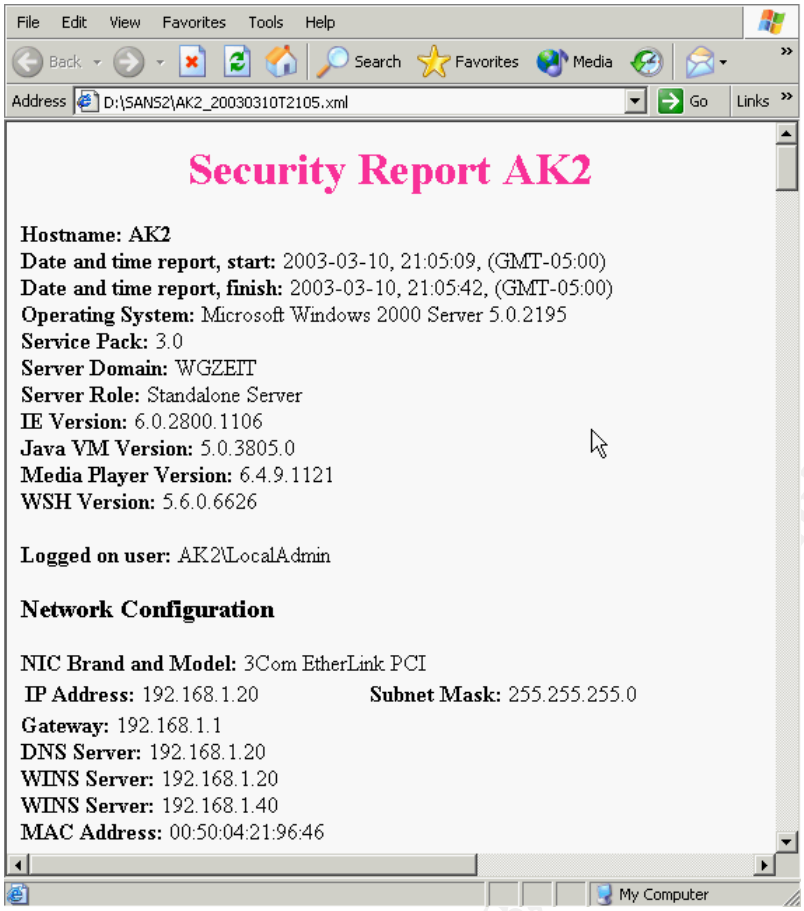
Execution time: varies from system to system and depends of both hardware performance and complexity of configuration. For typical Pentium-III class server with IIS installed it takes about 20 sec. to complete report – if report is directed on hard drive or network partition. Time approximately doubles if 1.44 MB floppy disk is used as media for report.

Validity verification for information collected with tool SecReport

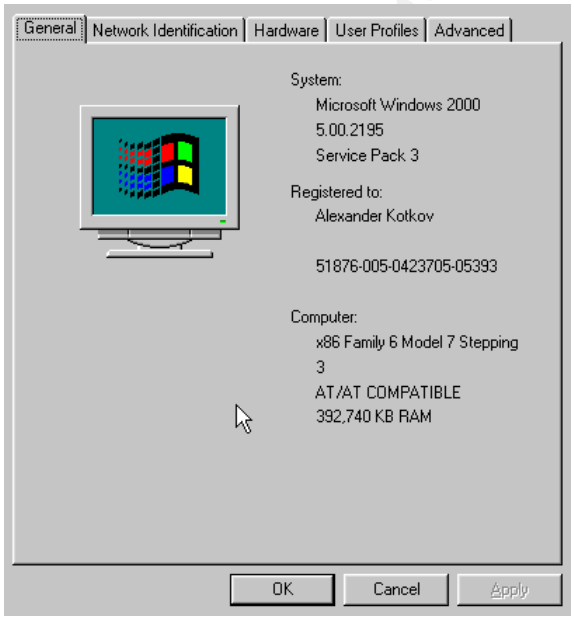
Due to wide spectrum of information collected by tool SecReport, I used variety of tools and techniques to verify validity of information, presented by report. General approach: where possible I used standard Windows OS tools (GUI or command line) to cross-check information in report. In situations where native Windows tools do not provide information that is verified, reputable third-party tools were used.

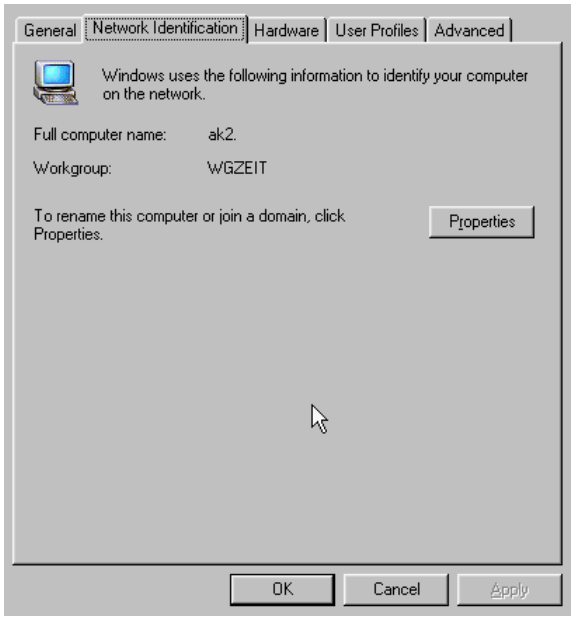
Verification of Hostname, Operating system, Service Pack, Server domain, CPU type and speed, RAM size.

On screenshot below presented fragment of report, generated by SecReport (full report was placed due to size considerations in Appendix 2.1)
This section of report contains general information about Windows system and network configuration.



This information was verified by checking data provided by Windows GUI and Windows command line tool ipconfig and Windows Resource Kit tool whoami.



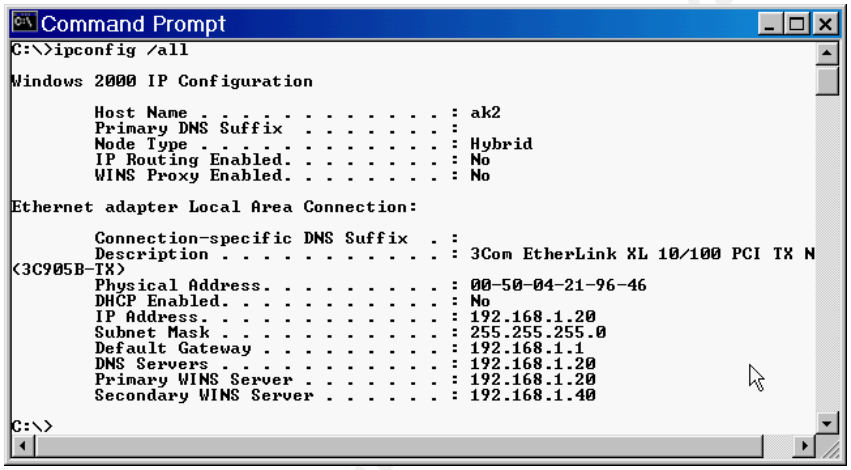


My computer – Right-click – Properties – Tabs General and Network Identification

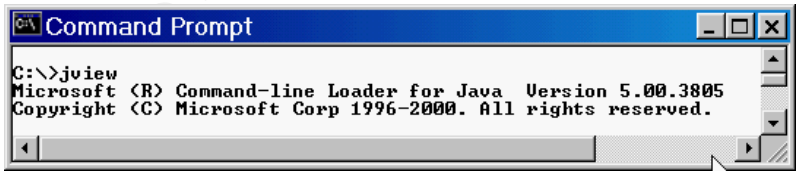
Information, obtained from this GUI tools positively verifies:

- Hostname;
- Server domain (listed as Workgroup *WGZEIT*, because server does not belong to domain, but configured as workgroup member);
- Operating system;
- Service pack of operating system;
- Type of CPU;
- Speed of CPU;
- Amount of RAM.

Screenshot of output of command line tool ipconfig confirms Network Configuration section of report.
Command whoami verifies logged on user.

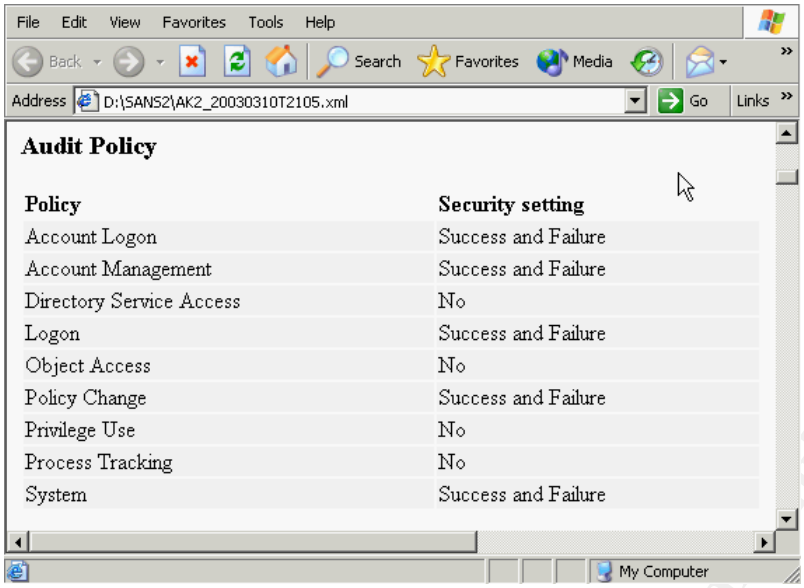


Version of Java Virtual Machine (JVM) was compared with output of command jview (this way is recommended by Microsoft in article: MSDN - Microsoft Visual J# .NET - The New Microsoft Java Virtual Machine[29] that can be found at: http://msdn.microsoft.com/vjsharp/productinfo/visualj/downloads/wfcinfo.asp).



Verification of Audit Policy section of report

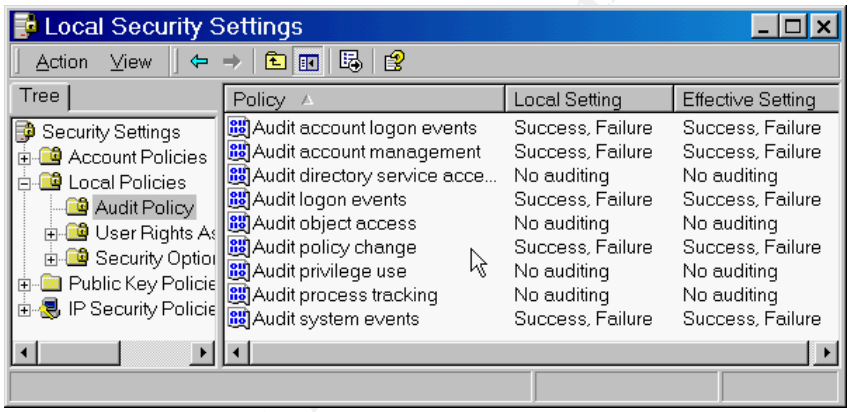
Screenshot below shows Audit Policy section of report.



Audit Policy

Policy	Security setting
Account Logon	Success and Failure
Account Management	Success and Failure
Directory Service Access	No
Logon	Success and Failure
Object Access	No
Policy Change	Success and Failure
Privilege Use	No
Process Tracking	No
System	Success and Failure

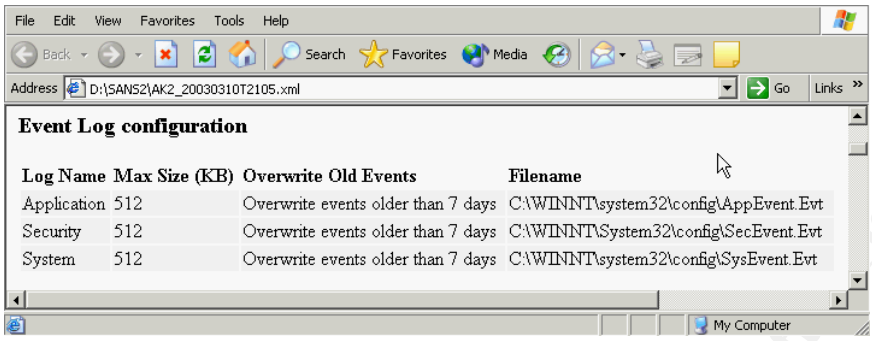
Validity of this data was verified by Windows GUI tool Local Security Settings (Start – Programs – Administrative Tools – Local Security Policy – Local Policies – Audit Policy), as presented on screenshot below:



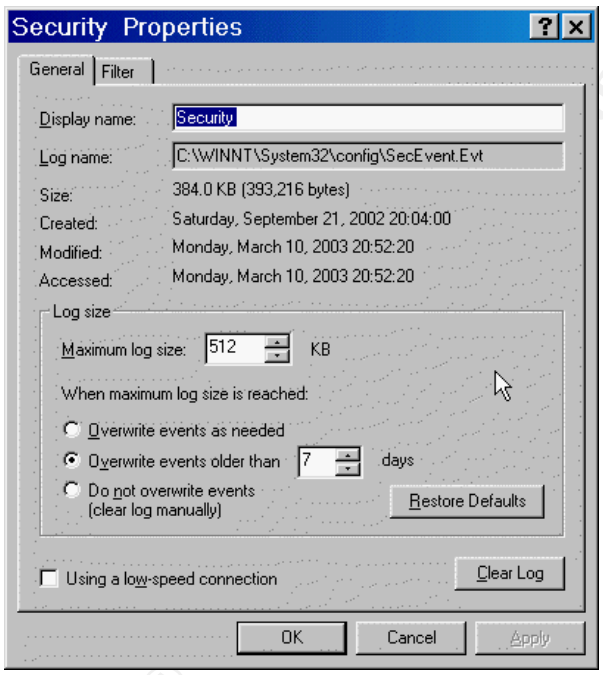
Data in Audit Policy section of report was identical to information obtained from Windows GUI tool.

Verification of Event Log configuration section of report

Event Log section of report was verified by Windows GUI tool Event Viewer (My Computer – Right-click – Manage – Event Viewer).

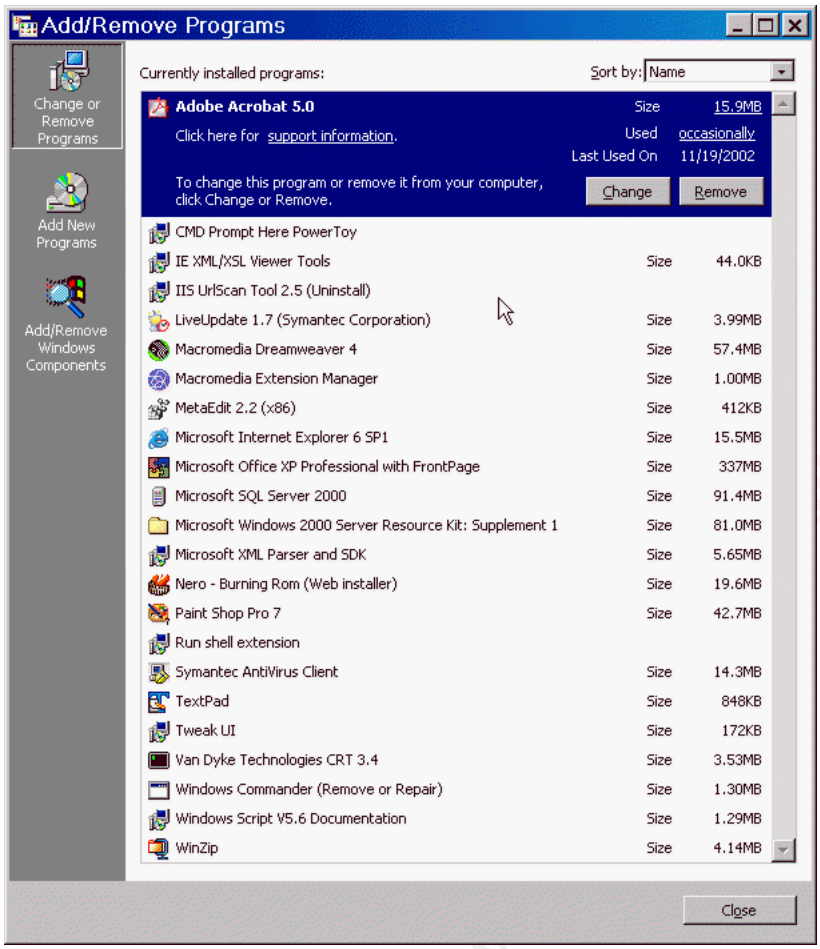


Screenshot of Event Viewer confirms correctness of Event Log Configuration section of report. Presented screenshot for Security log; for other logs (Application, System) results were also consistent between report and GUI tool.



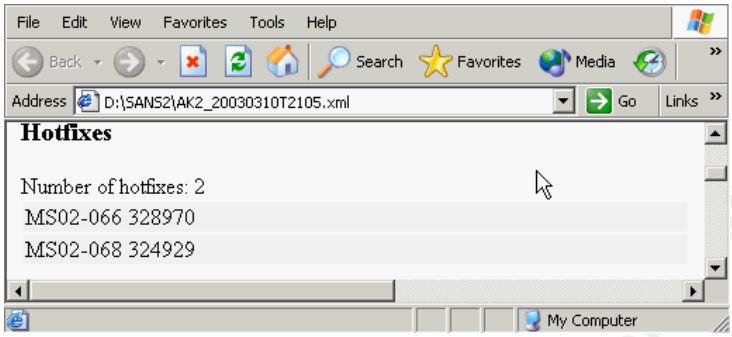
Verification of Applications section of report

Applications section of report was compared to list of installed applications, provided by Control Panel applet Add-Remove Programs (Start – Settings – Control Panel – Add/Remove Programs. This part of report was verified positively.



Verification of Hotfixes section of report

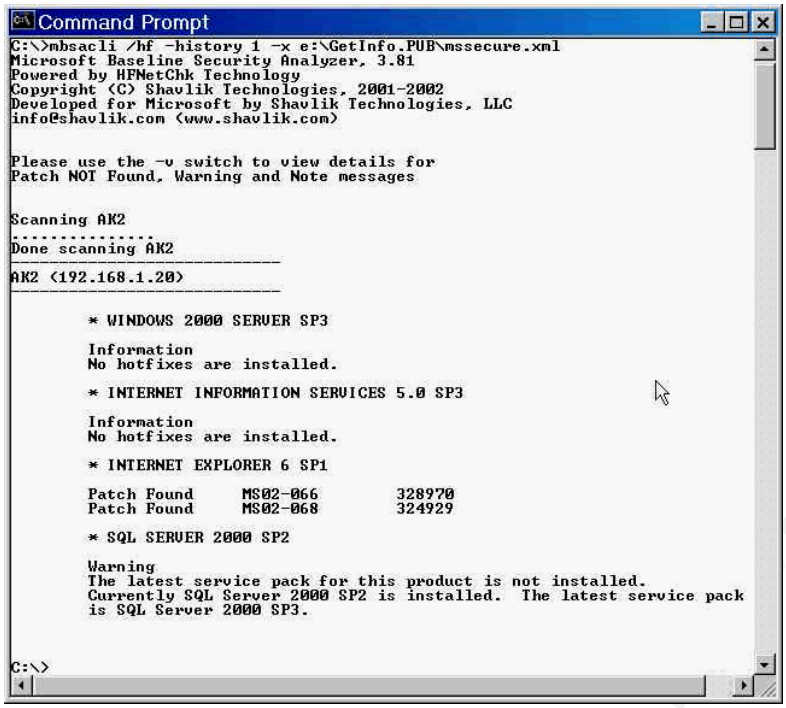
For verification of section Hotfixes I used command line tool Microsoft Baseline Security Analyser (MBSA) in "hfnetchk" mode (name of mode originates from name of previous version of software). In fact, same software – mbsacli.exe (and two auxiliary files for this program: hfdll.dll and mssecure.xml) are supplied as part of evaluated suite of tool and executed (as spanned process) in order to collect data on installed Microsoft hotfixes for Operating system (OS) and core components, tightly integrated with OS (IIS, IE, Media Player), also for few Microsoft Server products, such as MS SQL Server.



I executed command:

```
Mbsacli /hf -history 1 -x -e:\GetInfo.Pub\mssecure.xml
```

Parameter /hf intended to run mbsacli.exe in "hfnetchk" mode (mode of collecting information on installed hotfixes);
Parameter –history 1 means "show only explicitly installed hotfixes";
Parameter –x –e:\GetInfo.Pub\mssecure.xml specifies location of file with information on hotfixes. If this parameter is not specified, program mbsacli tries (without prompt) to connect to Microsoft web site in order to download latest version of file mssecure.xml (compressed as .cab archive). For forensic purposes it is recommended specifying location of local copy of file mssecure.xml – this prevents program from attempt to establish internet connection.

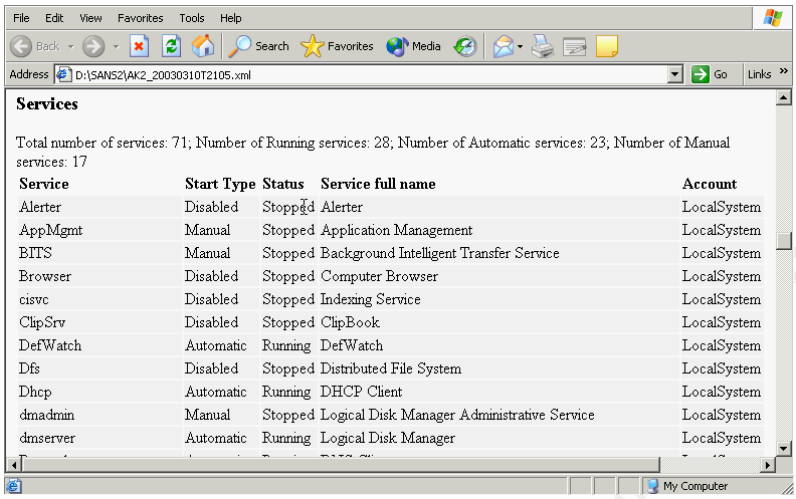


```
Command Prompt
C:\>mbsacli /hf -history 1 -x e:\GetInfo.PUB\mssecure.xml
Microsoft Baseline Security Analyzer, 3.81
Powered by HFNetChk Technology
Copyright (C) Shavlik Technologies, 2001-2002
Developed for Microsoft by Shavlik Technologies, LLC
info@shavlik.com (www.shavlik.com)


Please use the -v switch to view details for
Patch NOT Found, Warning and Note messages


Scanning AK2
................
Done scanning AK2
-----------------------------
AK2 (192.168.1.20)
-----------------------------

        * WINDOWS 2000 SERVER SP3

        Information
        No hotfixes are installed.

        * INTERNET INFORMATION SERVICES 5.0 SP3

        Information
        No hotfixes are installed.

        * INTERNET EXPLORER 6 SP1

        Patch Found     MS02-066        328970
        Patch Found     MS02-068        324929

        * SQL SERVER 2000 SP2

        Warning
        The latest service pack for this product is not installed.
        Currently SQL Server 2000 SP2 is installed.  The latest service pack
        is SQL Server 2000 SP3.


C:\>
```

Execution of mbsacli /hf command confirmed data, presented in Hotfixes section of report.

Verification of Services section of report

On screenshot below fragment of section Services presented (full report can be found in Appendix 2.1). Validity of information in this section was verified by Windows GUI program Services (Start – Programs – Administrative Tools – Services).

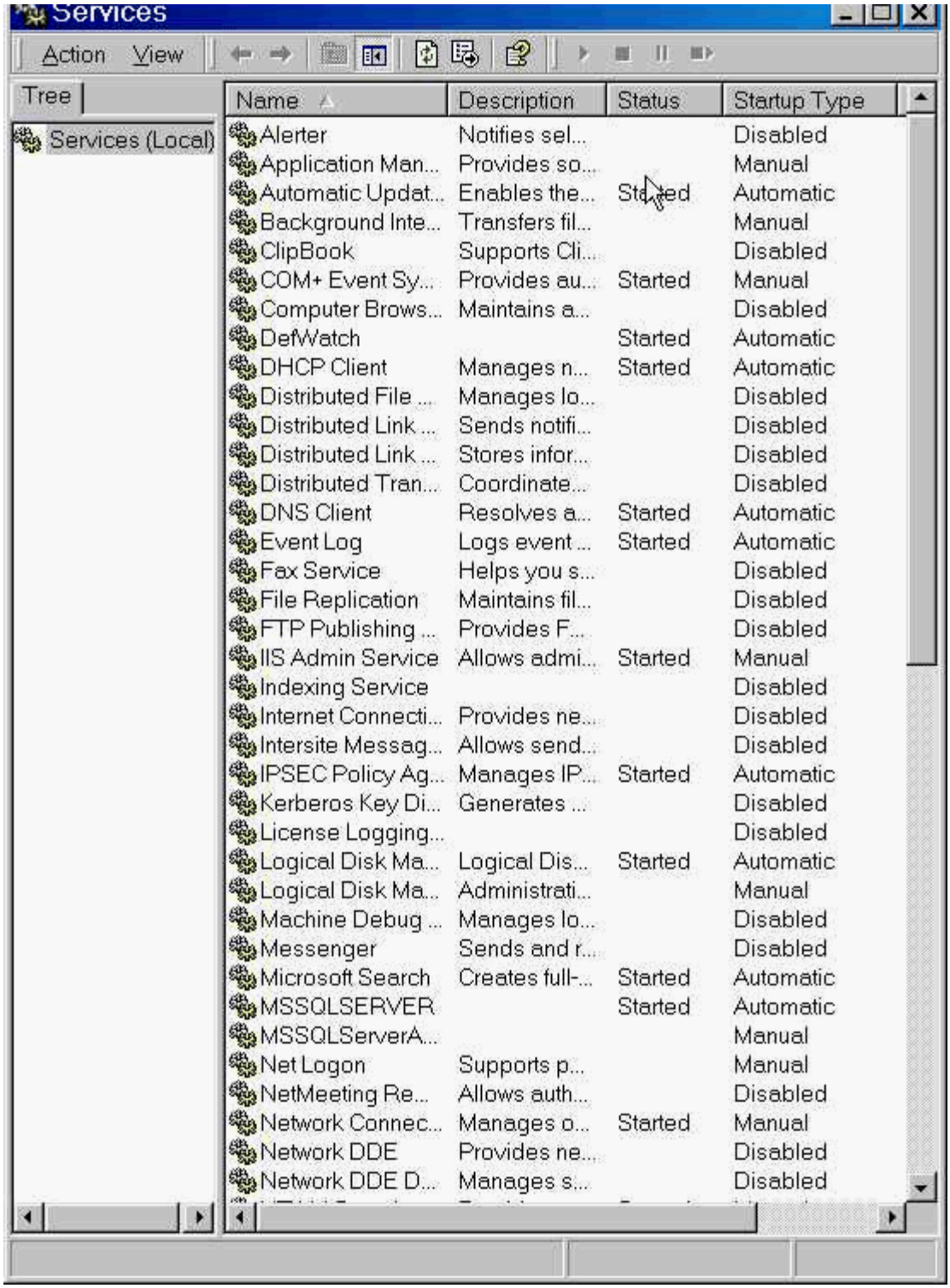


Services

Total number of services: 71; Number of Running services: 28; Number of Automatic services: 23; Number of Manual services: 17

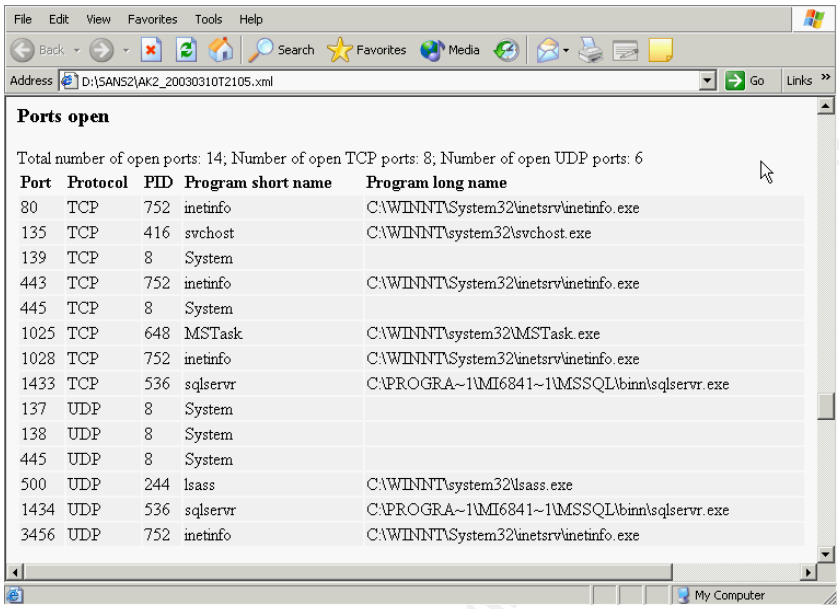
Service	Start Type	Status	Service full name	Account
Alerter	Disabled	Stopped	Alerter	LocalSystem
AppMgmt	Manual	Stopped	Application Management	LocalSystem
BITS	Manual	Stopped	Background Intelligent Transfer Service	LocalSystem
Browser	Disabled	Stopped	Computer Browser	LocalSystem
cisvc	Disabled	Stopped	Indexing Service	LocalSystem
ClipSrv	Disabled	Stopped	ClipBook	LocalSystem
DefWatch	Automatic	Running	DefWatch	LocalSystem
Dfs	Disabled	Stopped	Distributed File System	LocalSystem
Dhcp	Automatic	Running	DHCP Client	LocalSystem
dmadmin	Manual	Stopped	Logical Disk Manager Administrative Service	LocalSystem
dmserver	Automatic	Running	Logical Disk Manager	LocalSystem

Comparison of data from report to information, provided by Windows GUI tool Services found no discrepancies in tool's data. Below provided screenshot of part of information, presented by Windows GUI tool Services.



Verification of Ports Open section of report

In order to obtain data for this section of report, program SecReport starts (as spanned process) command line tool Fport.Exe v.2.0 [15] and parses its output. This free tool can be downloaded from: http://www.foundstone.com/knowledge/intrusion_detection.html and is considered to be reliable tool for this task. Fport.exe is provided together with evaluated set of tools.



Ports open

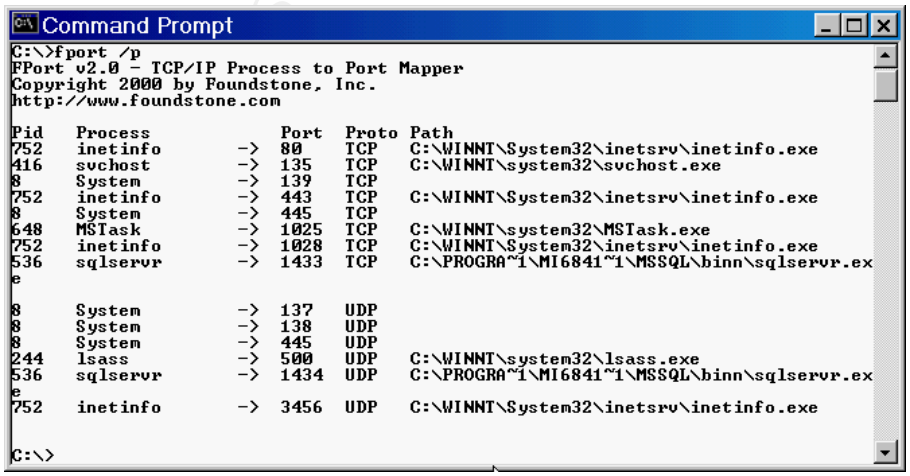
Total number of open ports: 14; Number of open TCP ports: 8; Number of open UDP ports: 6

Port	Protocol	PID	Program short name	Program long name
80	TCP	752	inetinfo	C:\WINNT\System32\inetsrv\inetinfo.exe
135	TCP	416	svchost	C:\WINNT\system32\svchost.exe
139	TCP	8	System	
443	TCP	752	inetinfo	C:\WINNT\System32\inetsrv\inetinfo.exe
445	TCP	8	System	
1025	TCP	648	MSTask	C:\WINNT\system32\MSTask.exe
1028	TCP	752	inetinfo	C:\WINNT\System32\inetsrv\inetinfo.exe
1433	TCP	536	sqlservr	C:\PROGRA~1\MI6841~1\MSSQL\binn\sqlservr.exe
137	UDP	8	System	
138	UDP	8	System	
445	UDP	8	System	
500	UDP	244	lsass	C:\WINNT\system32\lsass.exe
1434	UDP	536	sqlservr	C:\PROGRA~1\MI6841~1\MSSQL\binn\sqlservr.exe
3456	UDP	752	inetinfo	C:\WINNT\System32\inetsrv\inetinfo.exe

Execution of command:

```
Fport /p
```

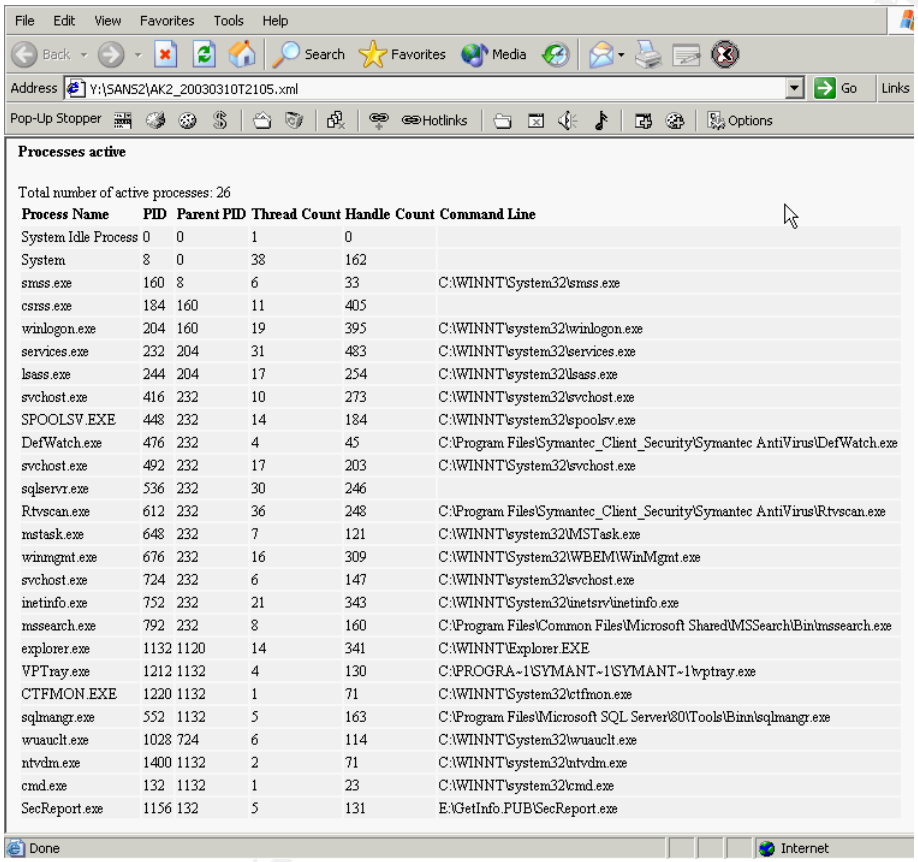
confirmed data, presented in report. Parameter /p specifies sorting order by port number.



```
C:\>fport /p
FPort v2.0 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com

Pid   Process          Port  Proto Path
752   inetinfo     ->  80    TCP   C:\WINNT\System32\inetsrv\inetinfo.exe
416   svchost      ->  135   TCP   C:\WINNT\system32\svchost.exe
8     System       ->  139   TCP
752   inetinfo     ->  443   TCP   C:\WINNT\System32\inetsrv\inetinfo.exe
8     System       ->  445   TCP
648   MSTask       ->  1025  TCP   C:\WINNT\system32\MSTask.exe
752   inetinfo     ->  1028  TCP   C:\WINNT\System32\inetsrv\inetinfo.exe
536   sqlservr     ->  1433  TCP   C:\PROGRA~1\MI6841~1\MSSQL\binn\sqlservr.ex
e

8     System       ->  137   UDP
8     System       ->  138   UDP
8     System       ->  445   UDP
244   lsass        ->  500   UDP   C:\WINNT\system32\lsass.exe
536   sqlservr     ->  1434  UDP   C:\PROGRA~1\MI6841~1\MSSQL\binn\sqlservr.ex
e
752   inetinfo     ->  3456  UDP   C:\WINNT\System32\inetsrv\inetinfo.exe

C:\>
```

Verification of Processes Active section of report

For verification of section Processes active of report I used tool psList.exe[30], developed by Mark Russinovich. PsList is free tool and can be downloaded from: http://www.sysinternals.com/ntw2k/freeware/pslist.shtml .
Reports, generated by SecReport and PsInfo had some format differences (in terms of properties that are displayed for each process), but list of active processes, their order and properties that are collected in both reports (Process name, Process ID, Thread Count, Handle Count) were identical.



Processes active

Total number of active processes: 26

Process Name	PID	Parent PID	Thread Count	Handle Count	Command Line
System Idle Process	0	0	1	0	
System	8	0	38	162	
smss.exe	160	8	6	33	C:\WINNT\System32\smss.exe
csrss.exe	184	160	11	405	
winlogon.exe	204	160	19	395	C:\WINNT\system32\winlogon.exe
services.exe	232	204	31	483	C:\WINNT\system32\services.exe
lsass.exe	244	204	17	254	C:\WINNT\system32\lsass.exe
svchost.exe	416	232	10	273	C:\WINNT\system32\svchost.exe
SPOOLSV.EXE	448	232	14	184	C:\WINNT\system32\spoolsv.exe
DefWatch.exe	476	232	4	45	C:\Program Files\Symantec_Client_Security\Symantec AntiVirus\DefWatch.exe
svchost.exe	492	232	17	203	C:\WINNT\System32\svchost.exe
sqlservr.exe	536	232	30	246	
Rtvscan.exe	612	232	36	248	C:\Program Files\Symantec_Client_Security\Symantec AntiVirus\Rtvscan.exe
mstask.exe	648	232	7	121	C:\WINNT\system32\MSTask.exe
winmgmt.exe	676	232	16	309	C:\WINNT\System32\WBEM\WinMgmt.exe
svchost.exe	724	232	6	147	C:\WINNT\system32\svchost.exe
inetinfo.exe	752	232	21	343	C:\WINNT\System32\inetsrv\inetinfo.exe
mssearch.exe	792	232	8	160	C:\Program Files\Common Files\Microsoft Shared\MSSearch\Bin\mssearch.exe
explorer.exe	1132	1120	14	341	C:\WINNT\Explorer.EXE
VPTray.exe	1212	1132	4	130	C:\PROGRA~1\SYMANT~1\SYMANT~1\vptray.exe
CTFMON.EXE	1220	1132	1	71	C:\WINNT\System32\ctfmon.exe
sqlmangr.exe	552	1132	5	163	C:\Program Files\Microsoft SQL Server\80\Tools\Binn\sqlmangr.exe
wuauclt.exe	1028	724	6	114	C:\WINNT\System32\wuauclt.exe
ntvdm.exe	1400	1132	2	71	C:\WINNT\system32\ntvdm.exe
cmd.exe	132	1132	1	23	C:\WINNT\system32\cmd.exe
SecReport.exe	1156	132	5	131	E:\GetInfo.PUB\SecReport.exe

Report, generated by SecReport tool shows:

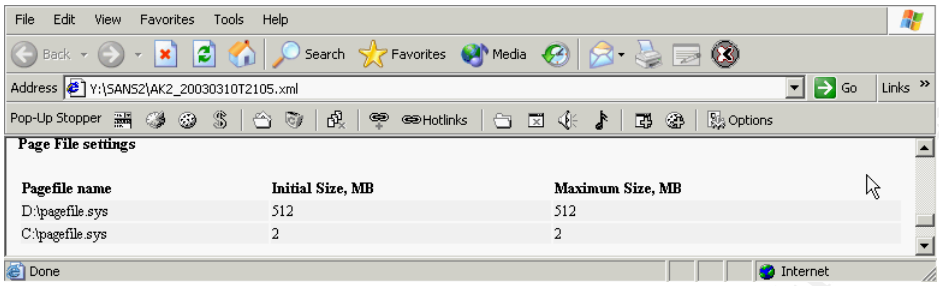
- Process name;
- Process ID;
- Process ID of parent process;
- Thread count;
- Handle count;
- Process ID of parent process (this information can be obtained from PsList by specifying command line parameter –t (for "show process tree");
- Command line that initiated process. I was not able to find tool that provides this information. Information about command line that initiated process collected according Microsoft specification: MSDN - Platform SDK: Windows Management Instrumentation - Win32_Process[31], that can be found at:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmisdk/wmi/win32_process.asp For some system processes this property is not provided by Windows OS.

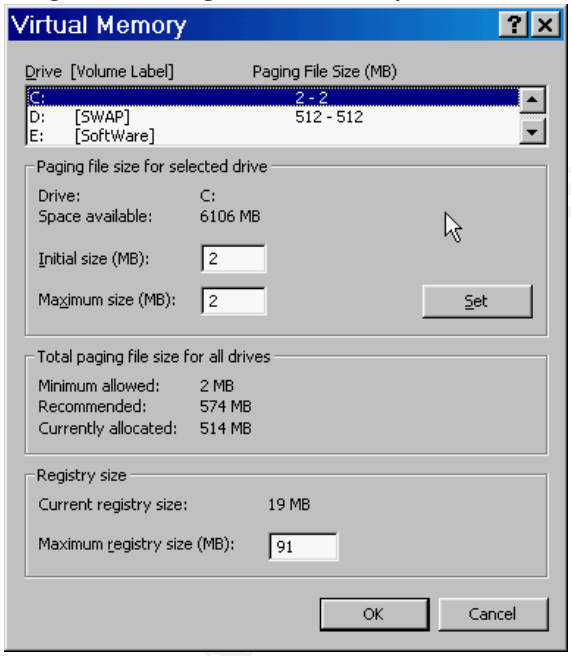
```
Command Prompt
C:\>pslist

PsList 1.21 - Process Information Lister
Copyright (C) 1999-2002 Mark Russinovich
Sysinternals - www.sysinternals.com

Process information for AK2:

Name          Pid Pri Thd  Hnd     Mem     User Time    Kernel Time    Elapsed Time
Idle            0   0   1    0      16   0:00:00.000   0:21:15.043    0:24:19.308
System          8   8  36  156     216   0:00:00.000   0:00:11.616    0:24:19.308
smss          160  11   6   33     336   0:00:00.010   0:00:00.751    0:24:19.308
csrss         184  13  11  374    2000   0:00:00.230   0:00:07.961    0:24:10.976
winlogon      204  13  18  391    3028   0:00:00.420   0:00:01.492    0:24:09.474
services      232   9  30  484    4880   0:00:01.321   0:00:03.495    0:24:07.781
lsass         244   9  17  254    4780   0:00:00.690   0:00:00.560    0:24:07.711
svchost       416   8  10  266    3500   0:00:00.110   0:00:00.250    0:24:03.055
SPOOLSV       448   8  11  138    3840   0:00:00.090   0:00:00.360    0:24:02.574
DefWatch      476   8   4   45    1692   0:00:00.030   0:00:00.040    0:24:02.444
svchost       492   8  14  196    5016   0:00:00.180   0:00:00.971    0:24:02.424
sqlservr      536   8  29  244   13852   0:00:00.190   0:00:00.380    0:24:00.501
Rtvscan       612   8  36  248    7608   0:00:00.380   0:00:03.174    0:23:52.159
mstask        648   8   7  121    2732   0:00:00.040   0:00:00.060    0:23:50.526
winmgmt       676   8   4  214     876   0:00:14.470   0:00:03.745    0:23:49.815
svchost       724   8   5  145    4784   0:00:00.040   0:00:00.160    0:23:49.225
inetinfo      752   8  18  331    6960   0:00:00.971   0:00:00.610    0:23:49.054
mssearch      792   8   7  158     952   0:00:00.110   0:00:00.260    0:23:48.544
explorer     1132   8  16  363    5304   0:00:06.218   0:00:19.658    0:23:31.399
VPTray       1212   8   3  128    3496   0:00:00.060   0:00:00.360    0:23:25.200
CTFMON       1220   8   1   74    2036   0:00:00.260   0:00:00.450    0:23:24.469
sqlmangr      552   8   4  161    4672   0:00:00.160   0:00:00.410    0:23:22.086
wuauclt      1028   8   4  109    4372   0:00:00.030   0:00:00.080    0:22:30.842
ntvdm        1400   8   2   71     940   0:00:43.502   0:00:18.556    0:21:40.459
cmd           924   8   1   23    1020   0:00:00.040   0:00:03.845    0:07:59.509
pslist       1416  13   2   84    1116   0:00:00.020   0:00:00.040    0:00:00.040

C:\>
```

Verification of Page File settings section of report

Data in section Page File settings was compared with data obtained from Windows GUI interface (Start – Settings – Control Panel – System – Advanced – Performance options – Virtual memory)



Data about Pagefile configuration obtained with SecReport were identical to data provided by Windows GUI interface. Screenshot of Windows 2000 GUI with Pagefile configuration data provided below.



Verification of data in Hardware section of report

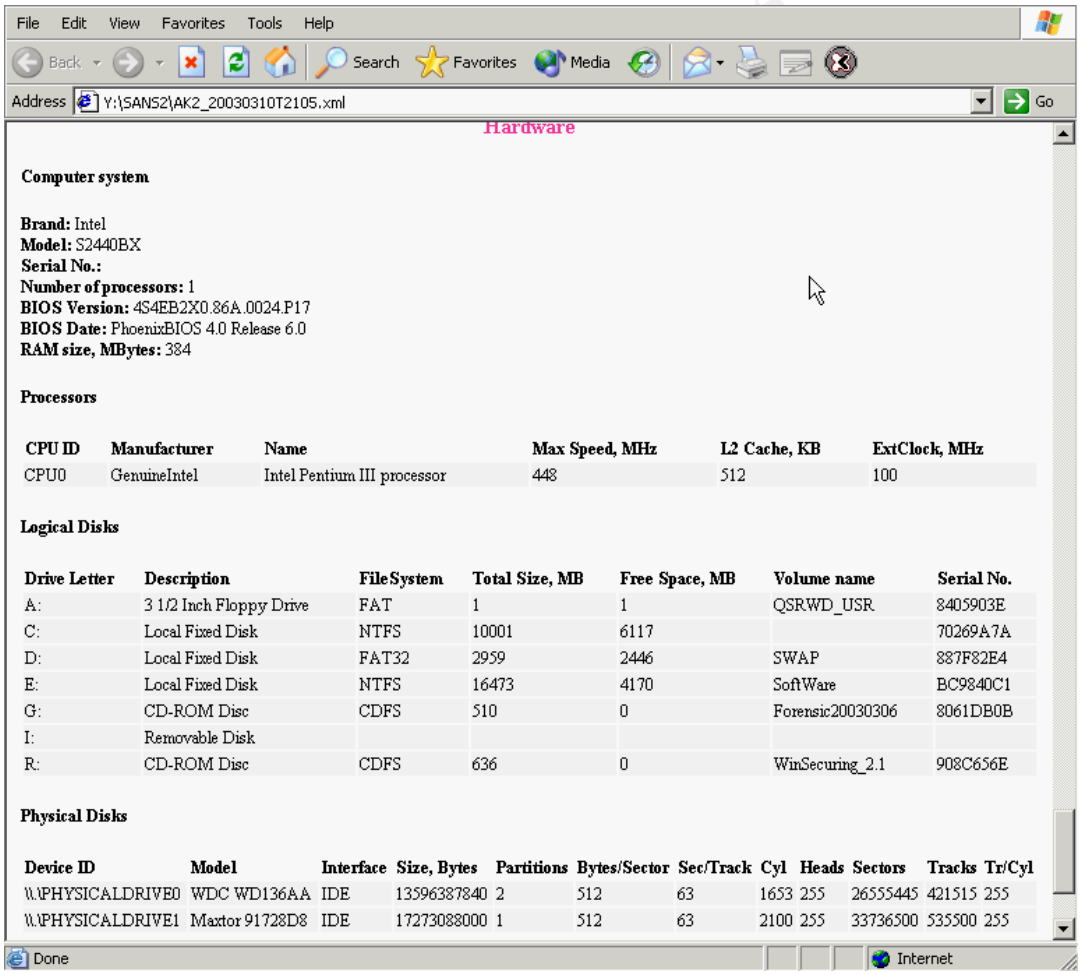
Data in this section was compared with:

- information provided by Windows GUI (see screenshots in section Verification of Hostname, Operating system, Service Pack, Server domain, CPU type and speed, RAM size);
- Disk Manager GUI tool of Windows OS;

and command-line tools:

- PsInfo, created by Mark Russinovich - http://www.sysinternals.com/ntw2k/freeware/psinfo.shtml

DiskMap – from Microsoft Windows 2000 Resource Kit. Tool can be downloaded at: http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/diskmap-o.asp



Hardware

Computer system

Brand: Intel
Model: S2440BX
Serial No.:
Number of processors: 1
BIOS Version: 4S4EB2X0.86A.0024.P17
BIOS Date: PhoenixBIOS 4.0 Release 6.0
RAM size, MBytes: 384

Processors

CPU ID	Manufacturer	Name	Max Speed, MHz	L2 Cache, KB	ExtClock, MHz
CPU0	GenuineIntel	Intel Pentium III processor	448	512	100

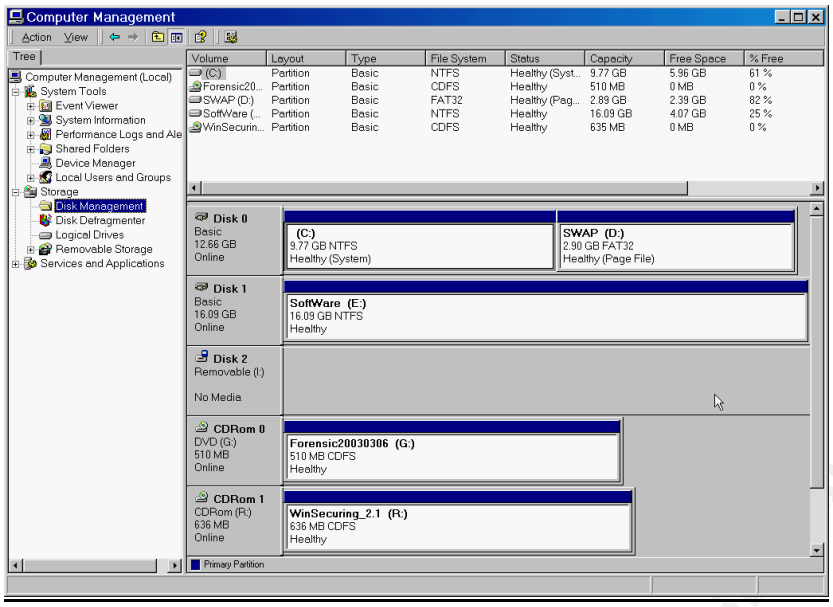
Logical Disks

Drive Letter	Description	File System	Total Size, MB	Free Space, MB	Volume name	Serial No.
A:	3 1/2 Inch Floppy Drive	FAT	1	1	QSRWD_USR	8405903E
C:	Local Fixed Disk	NTFS	10001	6117		70269A7A
D:	Local Fixed Disk	FAT32	2959	2446	SWAP	887F82E4
E:	Local Fixed Disk	NTFS	16473	4170	SoftWare	BC9840C1
G:	CD-ROM Disc	CDFS	510	0	Forensic20030306	8061DB0B
I:	Removable Disk					
R:	CD-ROM Disc	CDFS	636	0	WinSecuring_2.1	908C656E

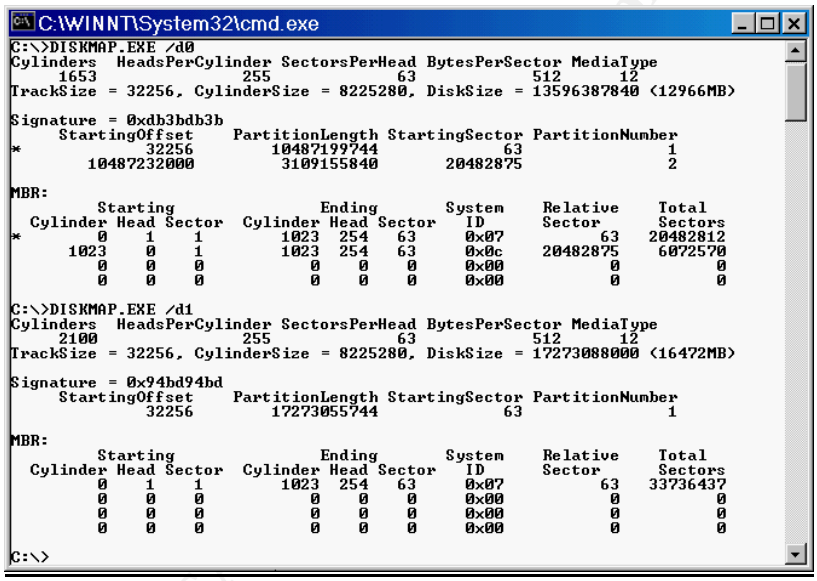
Physical Disks

Device ID	Model	Interface	Size, Bytes	Partitions	Bytes/Sector	Sec/Track	Cyl	Heads	Sectors	Tracks	Tr/Cyl
\\.\PHYSICALDRIVE0	WDC WD136AA	IDE	13596387840	2	512	63	1653	255	26555445	421515	255
\\.\PHYSICALDRIVE1	Maxtor 91728D8	IDE	17273088000	1	512	63	2100	255	33736500	535500	255

Screenshot of Hardware section of report



Screenshot of Disk Management Windows GUI tool.



```
C:\WINNT\System32\cmd.exe
C:\>DISKMAP.EXE /d0
Cylinders  HeadsPerCylinder SectorsPerHead BytesPerSector MediaType
     1653             255            63            512        12
TrackSize = 32256, CylinderSize = 8225280, DiskSize = 13596387840 (12966MB)

Signature = 0xdb3bdb3b
     StartingOffset     PartitionLength StartingSector PartitionNumber
*             32256         10487199744            63               1
        10487232000          3109155840      20482875               2

MBR:
          Starting                Ending        System     Relative     Total
  Cylinder Head Sector  Cylinder Head Sector    ID       Sector       Sectors
*         0    1    1       1023  254   63     0x07          63     20482812
      1023     0    1       1023  254   63     0x0c     20482875      6072570
         0    0    0          0    0    0     0x00            0            0
         0    0    0          0    0    0     0x00            0            0

C:\>DISKMAP.EXE /d1
Cylinders  HeadsPerCylinder SectorsPerHead BytesPerSector MediaType
     2100             255            63            512        12
TrackSize = 32256, CylinderSize = 8225280, DiskSize = 17273088000 (16472MB)

Signature = 0x94bd94bd
     StartingOffset     PartitionLength StartingSector PartitionNumber
             32256         17273055744            63               1

MBR:
          Starting                Ending        System     Relative     Total
  Cylinder Head Sector  Cylinder Head Sector    ID       Sector       Sectors
         0    1    1       1023  254   63     0x07          63     33736437
         0    0    0          0    0    0     0x00            0            0
         0    0    0          0    0    0     0x00            0            0
         0    0    0          0    0    0     0x00            0            0

C:\>
```

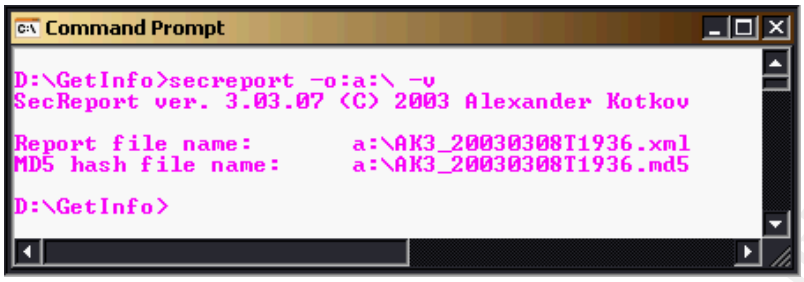
Screenshot of results from Diskmap tool. Tool was executed two times, once for every physical disk in system. Command line parameters: /d0 and /d1 specify number of physical disk.

Note: On some older (usually non-branded systems), manufactured before year 2000 some information (version of BIOS, , brand, model and serial number of computer) is not provided by OS and as result, not included in report.

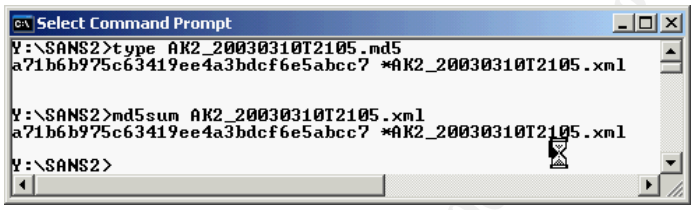
Results of verification for section Hardware were positive for all compared paramenters.

Verification of MD5 hash validity

At the end of execution tool SecReport generates MD5 hash of report file. MD5 hash placed into file with same filename as report, but has extension .md5. File with MD5 hash placed into same folder as report. SecReport shows location and filenames of both files during execution.



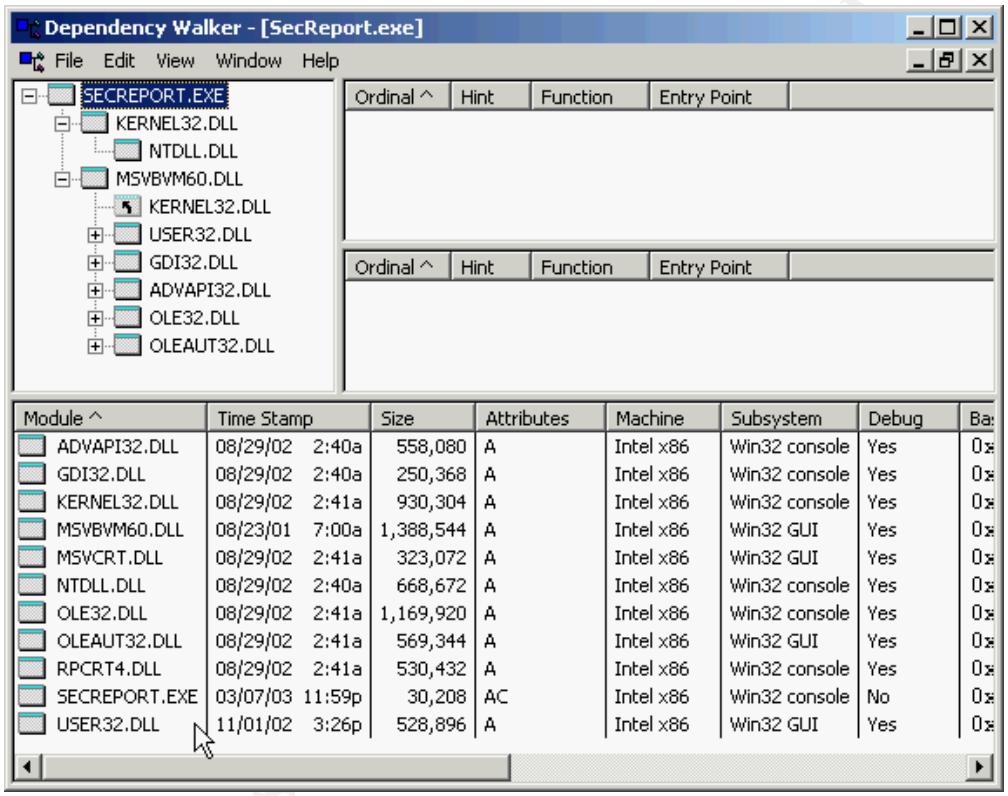
Validity of generated MD5 hash was validated by command line GNU tool md5sum[32] that was downloaded from http://www.gnu.org/software/textutils/textutils.html



Screenshot above indicates that MD5 hash, generated by SecReport is identical to MD5 hash value generated by command line tool md5sum.

How execution of tool SecReport changes status of target system

SecReport is command line tool that uses number of standard libraries of Windows OS. I used tool Dependency Walker[33] (tool is included in Microsoft Visual Studio 6 or can be downloaded from: http://www.dependencywalker.com/). This tool shows all libraries that are called from program. SecReport.exe directly calls just 2 Windows system libraries: kernel32.dll and msvbvm60.dll. These system libraries in turn call few other system libraries. Because program SecReport.exe uses number of system library files, it is suggested to include library files, listed in lower pane of screenshot into toolkit and use them from read-only media (such as CD-R) in order to avoid possibly trojaned binaries.



Screenshot of Dependency Walker program shows shared library files, used by SecReport.exe

To monitor files that were used during execution of SecReport tool I used tool FileMon[34] ver.5.00 created by Mark Russinovich. FileMon can be obtained from: http://www.sysinternals.com/ntw2k/source/filemon.shtml. Full results of running this tool are quite verbose and were placed in Appendix 2.2. In general, SecReport tool uses number of Windows system libraries and files (most of them are WSH and WMI-related). It is worth to note that there are no command line tools for Windows platform that do something meaningful and do not use any system library files.

Execution of tool does not modify any settings or parameters of target system and does not create any files on target system(if output and temporary files are redirected to network share or removable media).

In order to find out what system files have they MAC time changed as result of execution of SecReport tool, I did small test:

- Created small batch file testmac.bat with commands:

```
@echo on
date /t && time /t
secreport -o:a:\
@echo on
date /t && time /t
```

- After I executed batch file testmac.bat I noted date and time of start and end of SecReport execution: both times were: 2/5/2003, 13:12
- I run program macmatch[35] (it was created by Arne Vidstrom and can be found at his web site: http://www.ntsecurity.nu/toolbox/macmatch/). This program can detect all files accessed (when executed with parameter `-a`), modified (parameter -m) or created (parameter -c) within specified time period. According to results of this program, no files were created in C:\Winnt\ directory and its subdirectories; three files were accessed and modified:
    - C:\Winnt\System32\Config\Software
    - C:\Winnt\System32\Config\Software.log
    - C:\Winnt\System32\Perflib_Perfdata_33c.dat

```
Command Prompt
D:\GetInfo>testmac

D:\GetInfo>date /t   && time /t
Wed 02/05/2003
13:12

D:\GetInfo>call secreport -o:a:\ -t:a:\
SecReport ver. 3.02.03 (C) 2003 Alexander Kotkov

This program takes few minutes to complete ...

Report file name: a:\NY-KOTKOV2_20030205.xml

Program Completed !

D:\GetInfo>date /t   && time /t
Wed 02/05/2003
13:12

D:\GetInfo>macmatch c:\winnt\ -m 2003-02-05:13.12 2003-02-05:13.13

macMatch 1.0 - (c) 2002, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
             - http://ntsecurity.nu/toolbox/macmatch/

 c:\winnt\system32\config\software

   - M: 2003-2-5:13.12
   - A: 2003-2-5:13.12
   - C: 2002-10-24:15.23

 c:\winnt\system32\config\software.LOG

   - M: 2003-2-5:13.12
   - A: 2003-2-5:13.12
   - C: 2002-10-24:10.48

 c:\winnt\system32\Perflib_Perfdata_33c.dat

   - M: 2003-2-5:13.12
   - A: 2003-2-5:13.12
   - C: 2003-2-5:12.52


D:\GetInfo>macmatch c:\winnt\ -a 2003-02-05:13.12 2003-02-05:13.13

macMatch 1.0 - (c) 2002, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
             - http://ntsecurity.nu/toolbox/macmatch/

 c:\winnt\system32\config\software

   - M: 2003-2-5:13.12
   - A: 2003-2-5:13.12
   - C: 2002-10-24:15.23

 c:\winnt\system32\config\software.LOG

   - M: 2003-2-5:13.12
   - A: 2003-2-5:13.12
   - C: 2002-10-24:10.48

 c:\winnt\system32\Perflib_Perfdata_33c.dat

   - M: 2003-2-5:13.12
   - A: 2003-2-5:13.12
   - C: 2003-2-5:12.52


D:\GetInfo>macmatch c:\winnt\ -c 2003-02-05:13.12 2003-02-05:13.13

macMatch 1.0 - (c) 2002, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
             - http://ntsecurity.nu/toolbox/macmatch/


D:\GetInfo>
```

Because of this, it is highly suggested during incident response to use this tool after conducting MAC time report of file system, or (which is preferable, but not always practically feasible – after creating complete image of hard disk with dd or commercial imaging tools).

Recommendations on proper usage of tool SecReport for forensic purposes

Tool SecReport is single-file executable. Tool was written in Visual Basic 6 programming language and as result it uses VB6 shared library file msvbvm60.dll. This file supplied with default installations of Windows 2000, Windows XP and Windows 2003 (RC2) - located in folder %WinDir%\System32\. SecReport also spans three command line processes - auditpol, fport, mbsacli – in order to obtain some categories of information. Because of this, it is highly recommended to use following guidelines for usage of SecReport in situations of incident response and for forensic investigation:

- Place all files, provided in download archive getinfo.zip (can be downloaded from: http://kotkov.tripod.com/getinfo.zip or http://members.verizon.net/~vze3vkmg/tools/getinfo.zip) on read-only media (such as CD-R disk);
- Place to the same media files: msvbvm60.dll; cmd.exe. Cmd.exe file is OS-specific file, that need to be obtained from "clean" system.
- Redirect program output to removable media (floppy or Zip drive), using command line parameter: -o:pathname, where pathname is valid pathname of existing folder.
- Tool results are "sandwiched" between timestamps of start and finish of tool's execution. That allows to make necessary conclusions about scope of changed MAC (Modification, Access, Creation) times. To be on "safe" side it is recommended to run tool SecReport after MAC analysis.

Tool Delta

Tool Delta allows comparing any two reports, collected by tool SecReport. Reports can be collected from different systems – this is convenient in corporate environment, where most of computers are templated (by imaging software, such as Symantec Ghost [36] or PowerQuest DriveImage [37]) and generally, supposed to be identical. Running SecReport and Delta tools will indicate any differences in configuration (covered by tools) between two compared systems.
Applications for this type of comparison:

- Finding discrepancies in computer configurations can help detect rogue processes and applications installed on system;
- Data on hotfixes and detailed version information of major Windows components (OS, IE, Media player, Java VM, WSH etc) can allow fast assessment of potential vulnerability, that was exploited – it is based on fact that hotfixes prevent certain types of exploits. Same approach can be used in evaluating security status of IIS.

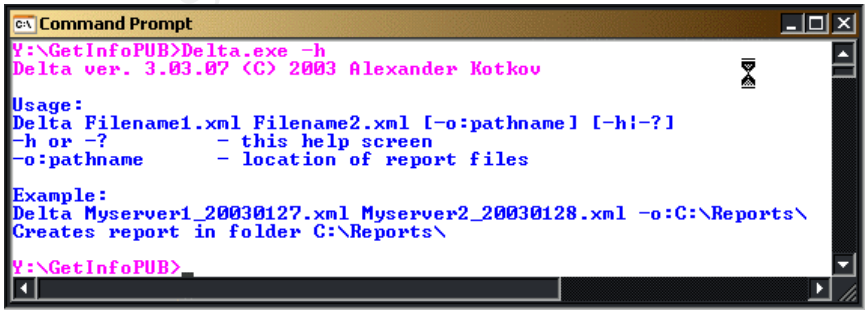
These tools were practically used in medium-size corporation for few months and were helpful in detection of few instances of unsanctioned remote access software ("backdoors").

Another scenario – creating reports from same system, but at different moments – that is useful in situation when current state of computer need to be compared to certain baseline.
Possible applications for this:

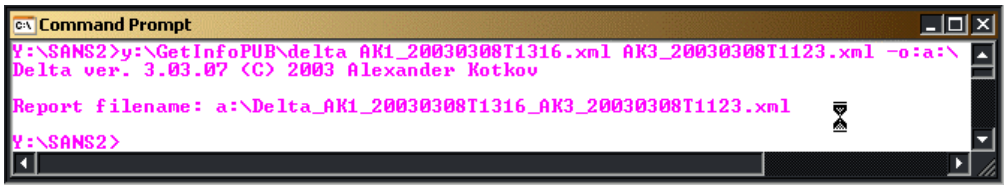
- honeynet installations;
- analysis of compromised system.

Usage of tool Delta

Tool Delta requires two mandatory parameters – filenames of reports to be compared. Optional parameter `-o:pathname` allows specify location for report files. If this parameter is not specified, tool places report files into root of %SystemDrive% (usually it means C:\ ). Program prints short help screen if typed with parameter –h or -? or without any parameters.



Screenshot of Help screen for program Delta

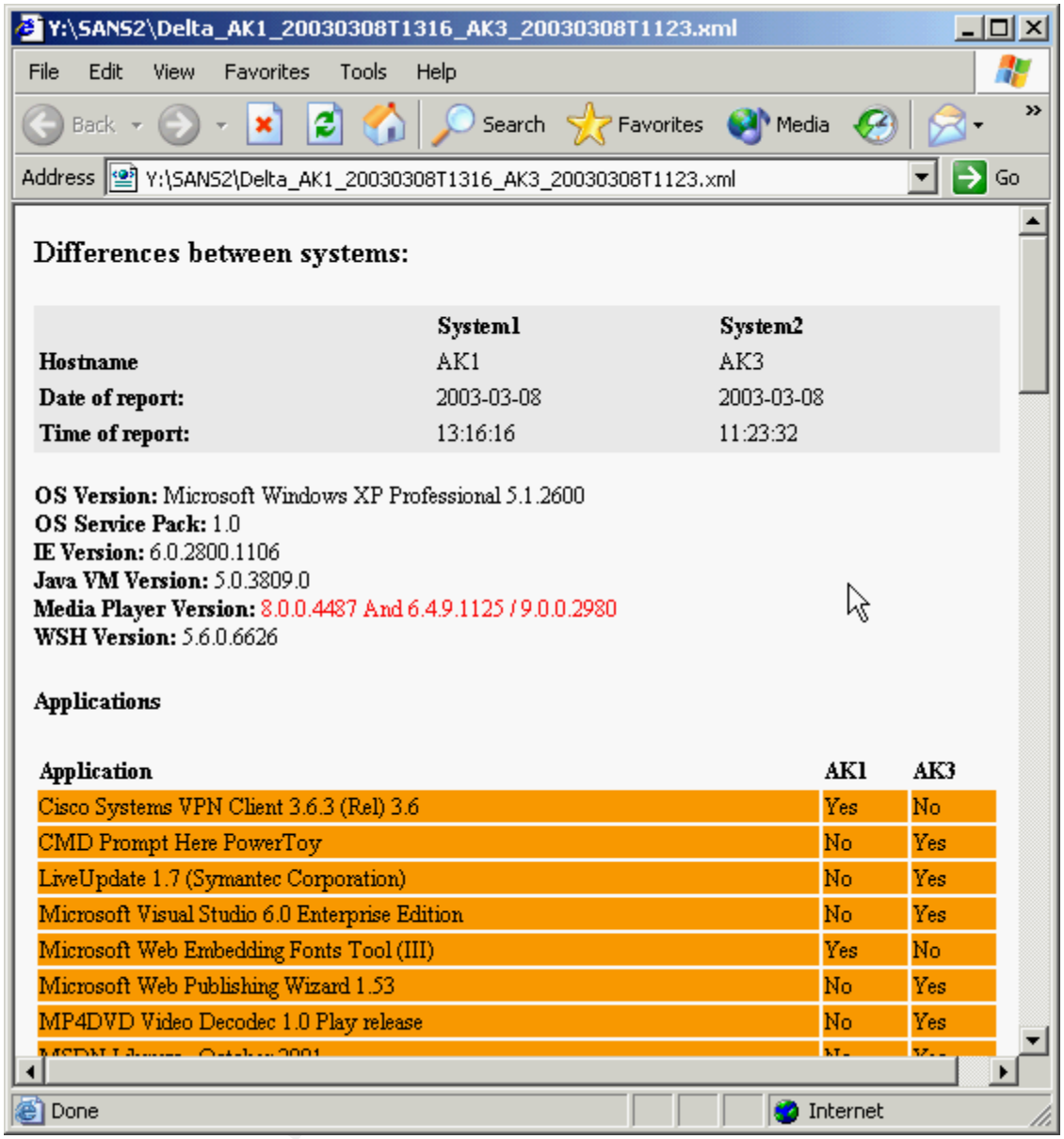


Screenshot of Delta program execution. Command line parameter: -o:a:\ in this example redirects output file to folder: a:\

Tool creates 2 report files:

- Report itself with filename created by template: Delta_Computername1YYYYMMDDHHM1_Computername2_yyyymmddhhm2.xml
  Where:
  - Computername1 and Computename2 – names of computers compared;
  - YYYY and yyyy – year of running SecReport tool on first and second systems, respectively;
  - MM and mm – month of running SecReport tool on first and second systems, respectively;
  - DD and dd – day of running SecReport tool on first and second systems, respectively;
  - HH and hh – hour of running SecReport tool on first and second systems, respectively – in "military" format: 0-23;
  - M1 and m2 – minute of running SecReport tool on first and second systems, respectively – 0-59;
- Auxiliary file DeltaRep.xsl - it is required for proper viewing and printing of reports in Internet Explorer 6 (this file is the same for all reports).

Execution time for tool Delta – usually about 3-5 seconds. It takes approximately 6-10 seconds to complete tool in case output is redirected to floppy disk.



Y:\SANS2\Delta_AK1_20030308T1316_AK3_20030308T1123.xml

File Edit View Favorites Tools Help

Address Y:\SANS2\Delta_AK1_20030308T1316_AK3_20030308T1123.xml

Differences between systems:

	System1	System2
Hostname	AK1	AK3
Date of report:	2003-03-08	2003-03-08
Time of report:	13:16:16	11:23:32

OS Version: Microsoft Windows XP Professional 5.1.2600
OS Service Pack: 1.0
IE Version: 6.0.2800.1106
Java VM Version: 5.0.3809.0
Media Player Version: 8.0.0.4487 And 6.4.9.1125 / 9.0.0.2980
WSH Version: 5.6.0.6626

Applications

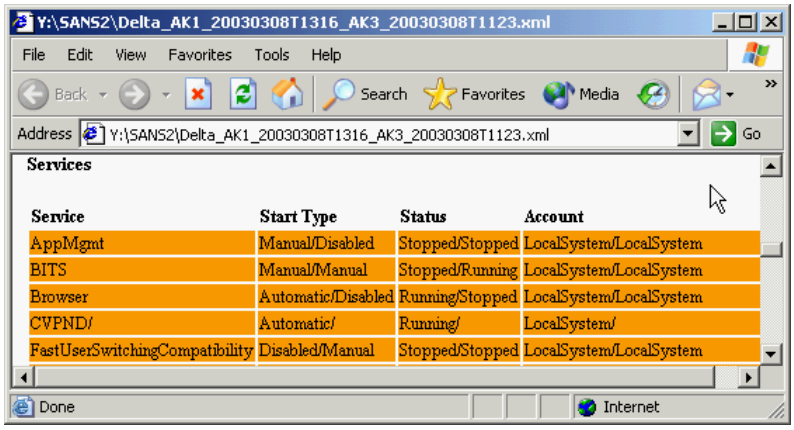
Application	AK1	AK3
Cisco Systems VPN Client 3.6.3 (Rel) 3.6	Yes	No
CMD Prompt Here PowerToy	No	Yes
LiveUpdate 1.7 (Symantec Corporation)	No	Yes
Microsoft Visual Studio 6.0 Enterprise Edition	No	Yes
Microsoft Web Embedding Fonts Tool (III)	Yes	No
Microsoft Web Publishing Wizard 1.53	No	Yes
MP4DVD Video Decodec 1.0 Play release	No	Yes

Done Internet

Screenshot of report (fragment) generated by tool Delta. Report viewed (and can be printed) from IE 6.0 browser. Full report was placed in Appendix 2.3 – due to size considerations.

Notes:

- If some value printed without / (slash character), that means that values are identical for System1 and System2 (In this report, for example, values "OS Version" are identical for both systems).
- If some values printed with "/" slash between them, that means that value before slash is for System1, value after slash – for System2. In this case, values are different. In example below, Media Player Version is different for systems.



Service	Start Type	Status	Account
AppMgmt	Manual/Disabled	Stopped/Stopped	LocalSystem/LocalSystem
BITS	Manual/Manual	Stopped/Running	LocalSystem/LocalSystem
Browser	Automatic/Disabled	Running/Stopped	LocalSystem/LocalSystem
CVPND/	Automatic/	Running/	LocalSystem/
FastUserSwitchingCompatibility	Disabled/Manual	Stopped/Stopped	LocalSystem/LocalSystem

Screenshot of report (fragment) generated by tool Delta - section Services. Full report was placed in Appendix 2.3 – due to size considerations.

- If some value has syntax:

`Value1 / –` that means that value exists only for System1 and does not exist for System2. In example below (Table "Services", entry "lanmanserver") service lanmanserver (and it's properties) exist only for System1.

- If some value has syntax:

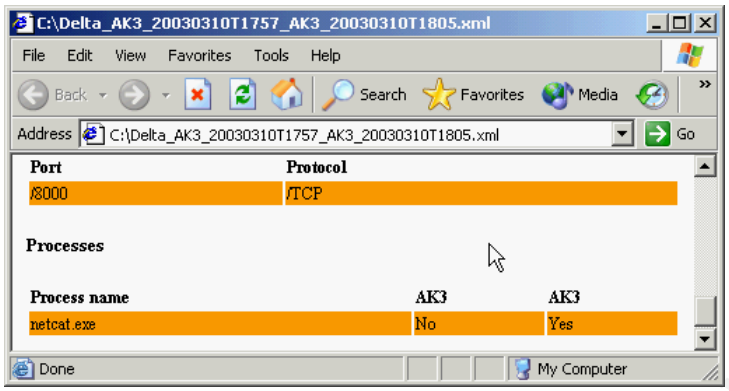
`/ Value2 –` that means that value exists only for System2 and does not exist for System1. In example below (Table "Services", entry "DefWatch") service DefWatch (and it's properties) exist only for System2.

How execution of tool Delta changes status of target system

Tool Delta does not need to be executed on any of target systems – it can be done on any Windows 2000 (or later) system. It does require IE 6.0 or later to be installed on system.
I monitored with earlier mentioned tool FileMon ver. 5.00 files that are affected by execution of this tool. There are very few system libraries, that this tool uses. With tool Process Explorer [38](created by Mark Russinovich from SysInternals - http://www.sysinternals.com/ntw2k/freeware/procexp.shtml) I monitored processes created by execution of Delta and memory footprint. Tool does not span any processes. Memory footprint is extremely low during execution of Delta.

Testing of repeatability and validity of reports, generated by tool Delta

Results of tool Delta were verified by running tool against random reports and subsequent manual verification of data validity. Results, provided by tool were correct in all tests.



Screenshot of report, generated by tool Delta after netcat tool was started in "listen" mode on tcp port 8000.

I also did test as follows:

- Run SecReport tool on standalone Windows 2000 Professional system – that generated "baseline" report;
- Disabled 2 services (Alerter and Messenger) on the system;
- Installed Adobe Acrobat Reader;
- Started netcat in "listen" mode – port 8000 as listener port;
- Run SecReport tool again on same PC – that generated "incident" report;
- Compared "baseline" and "incident" reports – all changes that were done to the system were properly detected.

Analysis

Tool SecReport collects information on variety of parameters of Windows system, that can be useful for forensic investigation. Full list of collected information is provided in section: Information, collected by tool SecReport of present paper. Briefly, main categories of information: versions of OS and OS components, network configuration, services, processes, applications, hotfixes, IIS configuration (if applicable), hardware specifications. This information is generally collected by forensic investigator from any system under investigation. Usually set of tools, scripts etc is used to collect same information. This process is labor-intensive and error-prone. Tool SecReport also attempts to present collected information in structured, clear way, that make it more understandable by non-technical audience. Detailed verification of tool, presented earlier, proves that tool provides reliable, repeatable and reproducible information that can withstand, if challenged in court. Reports, generated by tool are "self-describing" XML files. By using supplied template securityreport.xsl reports can be viewed and printed in "human-friendly" form from Internet Explorer 6.0.

Tool Delta serves as productivity tool in tasks of comparison of information, collected from any two systems (or same system at different moments of time). It does its job by automatically comparing number of parameters from different systems and presenting "difference" report. This is quite generic task, that is extremely time-consuming and error-prone if conducted manually.

Conclusion

Evaluation of tools SecReport and Delta as forensic tools makes me feel that these tools can be successfully used for forensic work.
These tools are:

- Detailed validation of results, obtained with tool SecReport, described in previous sections proved reliability and repeatability of results. This kind of validation allows cross-checking of results by alternative tools in case results are challenged in court.
- Monitoring of tool's impact on target system indicates that only very small number of system files change their Access time. No files are created, deleted or modified on target system.
- Tool automatically generates MD5 hash for report that it generates – that is standard requirement for data, collected during incident response.
- Raw report files are structured XML files – that makes them "self-describing" and "self-commenting".
- Reports, generated by SecReport can be viewed and printed from Microsoft Internet Explorer 6.0 or later browsers.
- Report is formatted the way that promotes understanding of results by non-technical audience (use of logically organized sections, fonts, color etc).

- Advantage of running SecReport over collecting information using standard Windows GUI tools (and some information from Windows can be obtained only through use of GUI) – significantly less impact on target system (in fact, monitoring process of just opening Control Panel with mentioned earlier tool FileMon shows that it involves much more activity and modifies files). Also, generally GUI tools do not provide standard functionality to capture displayed information in text file (in many cases, it is necessary to handwrite displayed information).
- Tool Delta allows to automate monotonous and error-prone, labor-intensive process of comparing data, collected from different systems (or at different times), which can be very valuable during initial incident response and subsequent analysis

Limitations of tools: support only for Windows 2000, XP and 2003 (RC-2) platforms – it is not a big disadvantage, taking into account that these versions of Windows are installed on millions of systems worldwide.

Tests that I conducted on these tools prove their functionality, reliability and repeatability.

All above-mentioned facts bring me to opinion that tool SecReport can be useful in arsenal of computer forensic investigator on all stages of investigation – from initial collection of information from Windows system to presenting results to law enforcement and in court.

Part 3 - Legal Issues of Incident Handling

Introduction

Presented scenario gives me, as system administrator of Internet Service Provider (ISP) legal rights to collect information that can provide help to law enforcement organizations and protect my rights and property.
There are also certain procedural limitations that are dictated by law.
What makes this scenario special and provides more flexibility both for law enforcement officers and for me, as system administrator of ISP – fact that hacker attacked government computer – by classification of applicable laws government computers belong to broader definition of "protected computers".
Also, changes in cyber crime related laws that occurred within last two years allow effective communication between law enforcement organizations and ISP and prevent unnecessary procedural delays (that hampered many similar cases in past).

Answers to Practical Assignment questions

A. In described scenario, not much information can be provided during initial phone conversation, even if identity of law enforcement officer was validated. Administrator of ISP can just give general contact information, such as his name, position etc. During this initial conversation, ISP administrator can request law enforcement officer to obtain:

- Preservation request letter - under United States Code (U.S.C.) §2703 (f)[39];
- Subpoena;
- Court order;
- Search warrant.

Any details, relevant to crime, cannot be divulged over the phone in described situation.

The only practical outcome of this initial phone conversation can be verbal request from law enforcement officer to preserve relevant information (but this is not recommended option) – according to Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations[40] Section III-G: "While a simple phone call should therefore be adequate, a fax or an e-mail is better practice because it both provides a paper record and guards against miscommunication. Upon receipt of the government's request, the provider must retain the records for 90 days, renewable for another 90-day period upon a government request."

B. In case of delay in obtaining required legal authority, law enforcement officer must provide Preservation Request Letter – according to U.S.C. §2703 (f), which mandates: "A provider of wire or electronic communication services or a remote computing service, upon request of a governmental entity, shall take all necessary steps to preserve records and other evidence in its possession pending the issuance of a court order or other process."
In case there will be further delay(s) in obtaining required legal authority,, U.S.C. §2703(f)(2) specifies that "Records referred to in paragraph (1)" (of U.S.C. §2703(f)) "shall be retained for a period of 90 days, which shall be extended for an additional 90-day period upon a renewed request by the governmental entity."

Document Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations in Appendix C[41] provides sample Preservation Request Letter. According to this sample, law enforcement officer can request preservation for 90 days:

- "All stored communications and other files reflecting communications to or from [Email Account / User name / IP Address or Domain Name (between DATE1 at TIME1 and DATE2 at TIME2)];
- All files that have been accessed by [Email Account / User name / IP Address or Domain Name (between DATE1 at TIME1 and DATE2 at TIME2)] or are controlled by user accounts associated with [Email Account / User name / IP Address or Domain Name (between DATE1 at TIME1 and DATE2 at TIME2)];
- All connection logs and records of user activity for [Email Account / User name / IP Address or Domain Name (between DATE1 at TIME1 and DATE2 at TIME2)], including;

    1. Connection date and time;
    2. Disconnect date and time;
    3. Method of connection (e.g., telnet, ftp, http);
    4. Type of connection (e.g., modem, cable / DSL, T1/LAN);
    5. Data transfer volume;
    6. User name associated with the connection and other connection information, including the Internet Protocol address of the source of the connection;
    7. Telephone caller identification records;
    8. Records of files or system attributes accessed, modified, or added by the user;
    9. Connection information for other computers to which the user of the [Email Account / User name / IP Address or Domain Name (between DATE1 at TIME1 and DATE2 at TIME2)] connected, by any means, during the connection period, including the destination IP address, connection time and date, disconnect time and date, method of connection to the destination computer, the identities (account and screen names) and subscriber information, if known, for any person or

entity to which such connection information relates, and all other information related to the connection from ISP or its subsidiaries."

As we can see, this letter request to preserve very broad spectrum of information, related to ISP and its subscribers.

C. Law officer need to provide me, as ISP system administrator, one of following documents in order to receive logs from equipment of ISP that I operate or own:

- Subpoena: According to Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations, Appendix E - Sample Subpoena Language there are few valid types of subpoena for this situation:
    - Administrative subpoena authorized by Federal or State statute;
    - Federal or State grand jury or
    - Trial subpoena;
- Court order – "issued by any court that is a court of competent jurisdiction" – according to U.S.C. §2703 (d);
- Search warrant. See U.S.C. § 2703(c)(2).

D. As ISP administrator I cannot conduct any investigative activity on my own. All investigations must be conducted by authorized personnel of law enforcement organizations.
There are few rules that I must follow in order to cooperate with investigators and provide necessary environment for successful investigation:

- Preserve necessary evidence according to Preservation Request Letter obtained from law enforcement officer – according to U.S.C. § 2703(f)(1). There is limitation: this section of law specifies preservation of already existing data; if future communications need to be monitored, there should be compliance with requirements for electronic surveillance.
- Not to disclose existence of subpoena, court order or search warrant to any person – this particular situation can fall under one or more of following:
    - Endangering the life or physical safety of an individual; - 18 U.S.C. §2705(b)(1)
    - flight from prosecution - 18 U.S.C. § 2705(b)(2);
    - destruction of or tampering with evidence prosecution - 18 U.S.C. §2705(b)(3);
    - intimidation of potential witnesses prosecution - 18 U.S.C. §2705(b)(4);
    - otherwise seriously jeopardizing an investigation or unduly delaying a trial prosecution - 18 U.S.C. § 2705(b)(5).

Unconditionally, I cannot retaliate hackers attack – that will alert attacker that his/her activity was disclosed and monitored and this can cause unrecoverable damage to investigation.

E. In case when ISP logs disclosed fact that hacker gained unauthorized access to my system, created an account for his/her use and use that account to hack into government system, in addition to measures that are required from me in order to cooperate with law enforcement, I obtain some additional rights. In fact, proven unauthorized access to ISP system qualifies as an offence - under U.S.C. §2701(a)[42]. In addition to information requested by law enforcement, ISP "may divulge a record or other information, pertaining to a subscriber or customer of such service … as may be necessary … to the rendition of the service or to the protection of the rights or property of the provider of that service".

Question (**E**) of Practical Assignment can be interpreted also other way – I discovered (by reviewing logs) that my ISP system was compromised and used as starting point for hacking into government computer, but I was not contacted by law enforcement officer. In this case, it is my responsibility to report this incident as soon as possible to appropriate law enforcement authorities (National Infrastructure Protection Center; FBI, US Secret Service, US Customs Service etc).[43]

Activity of hacker described in scenario qualifies as fraud under U.S.C § 1030 (a)(2-6)[44]. In addition to crime of hacking into government computer system, fact of obtaining unauthorized account on ISP and using it for hacking of other systems under U.S.C. §1030 (a)(1) qualifies as fraud and punishable with a fine or imprisonment up ten years (under U.S.C. §1030(c)(1)(A)) or up to twenty years (under U.S.C. §1030(c)(1)(B)) – depending of other convictions.

In case this incident would involve any party, located in state of New York, there additional considerations that can qualify this crime as computer tampering as first degree, because it meets the definition in McKinney's Consolidated Laws Of New York Annotated Penal Law. Chapter 40, Part Three, Title J, Article 156[45], which is applicable in described scenario for the reason that hacker used compromised ISP system in order to commit or attempt to commit furter tampering. Under this legislation this crime qualifies as class E felony.

Conclusion

For the last two years number of legislative acts and laws, related to cyber crime were introduced in the United States of America. There are few reasons for this:

- Tragic events of September 11th, 2001, when organized group of international terrorists hijacked four passenger airplanes and destroyed Twin Towers in New York and attacked Pentagon in Washington, DC, killing thousands of people. Investigation of this monstrous crime proved that terrorists extensively used modern communication and computer technologies in order to prepare and conduct this act of terror;

- Few high-profile crimes (trial of David Smith – author of Melissa virus[46]; case of Gorshkov & Ivanov - extortion of money[47]; Gary McKinnon – hacked numerous computers of US Army and NASA[48] etc) indicated that computer crime migrated from category of "exotic and rare" cases to which they belonged decade or more ago into mainstream of criminal activity;
- Changes in technology – some of the laws that regulate computer-related crime were originally created 30-50 years ago (Atomic Energy Act, 1954; Omnibus Crime Control and Safe Streets Act of 1968) and contain technology-specific language [appropriate for level of technology of that time], sometimes affected effectiveness of investigation and trial;
- Omnipresence of computer technologies in every segment of modern life and reliance of virtually any critical element of infrastructure on data processing and electronic communications.

Two fundamental legislative acts that affects number of laws dealing with cyber crimes - USA Patriot Act of 2001[49] and The Cyber Security Enhancement Act (part of Homeland Security Act of 2002)[50].

Sections 210-212, 216, 217, 220, 814-816 of USA Patriot Act of 2001 make number of amendments to:

- Electronic Communications Privacy Act (ECPA) – United States Code (U.S.C.) §§ 2701-2712,
- Computer Fraud and Abuse Act - U.S.C. §1030;
- Pen Register and Trap and Trace Statute - U.S.C. § 3121, 3123, 3124 and 3127.

These amendments have a goal of expanding scope of existing laws in order to provide coverage of existing and perspective communication and computer technologies, avoid ambiguity in interpreting these laws and provide more rights to law enforcement in order to facilitate persecution and prevention of computer related crimes.

Cyber Security Enhancement Act provides further enhancements to laws, related to computer crime. Namely, it amends following laws:[51]

- Title 18, Section 1030, Fraud and related activity in connection with computers
- Title 18, Section 2511, Interception and disclosure of wire, oral, or electronic communications prohibited
- Title 18, Section 2512, Manufacture, distribution, possession, and advertising of wire, oral, or electronic communication intercepting devices prohibited
- Title 18, Section 2517, Authorization for disclosure and use of intercepted wire, oral, or electronic communications
- Title 18, Section 2520, Recovery of civil damages authorized
- Title 18, Section 2701, Unlawful access to stored communications
- Title 18, Section 2702, Voluntary disclosure of customer communications or records

- Title 18, Section 2703, Required disclosure of customer communications or records
- Title 18, Section 3125 Emergency pen register and trap and trace device installation

This level of attention of government and legislators clearly indicates significance of contemporary, powerful and effective legislation for fighting with cyber crime.

Appendixes

Appendix 1.1. Output of zipinfo –v binary_v1.2.zip command

Note: Command zipinfo was executed with parameter –v – for verbose output.

```
# zipinfo -v binary_v1.2.zip
Archive:  binary_v1.2.zip   7309 bytes   2 files

End-of-central-directory record:
--------------------------------

  Actual offset of end-of-central-dir record:        7287 (00001C77h)
  Expected offset of end-of-central-dir record:       7287 (00001C77h)
  (based on the length of the central directory and its expected offset)

  This zipfile constitutes the sole disk of a single-part archive; its
  central directory contains 2 entries.  The central directory is 102
  (00000066h) bytes long, and its (expected) offset in bytes from the
  beginning of the zipfile is 7185 (00001C11h).

  There is no zipfile comment.

Central directory entry #1:
---------------------------

  atd.md5

  offset of local header from start of archive:     0 (00000000h) bytes
  file system or operating system of origin:        MS-DOS, OS/2 or NT FAT
  version of encoding software:                     2.0
  minimum file system compatibility required:       MS-DOS, OS/2 or NT FAT
  minimum software version required to extract:     2.0
  compression method:                               deflated
  compression sub-type (deflation):                 normal
  file security status:                             not encrypted
  extended local header:                            no
  file last modified on (DOS date/time):            2002 Aug 22 14:58:08
  32-bit CRC value (hex):                           e5376cb4
  compressed size:                                  38 bytes
  uncompressed size:                                39 bytes
  length of filename:                               7 characters
  length of extra field:                            0 bytes
  length of file comment:                           0 characters
```

```
  disk number on which file begins:                disk 1
  apparent file type:                              text
  non-MSDOS external file attributes:              81B600 hex
  MS-DOS file attributes (20 hex):                 arc

  There is no file comment.

Central directory entry #2:
---------------------------

  atd

  offset of local header from start of archive:    75 (0000004Bh) bytes
  file system or operating system of origin:       MS-DOS, OS/2 or NT FAT
  version of encoding software:                    2.0
  minimum file system compatibility required:      MS-DOS, OS/2 or NT FAT
  minimum software version required to extract:    2.0
  compression method:                              deflated
  compression sub-type (deflation):                normal
  file security status:                            not encrypted
  extended local header:                           no
  file last modified on (DOS date/time):           2002 Aug 22 14:57:54
  32-bit CRC value (hex):                          d0ee3072
  compressed size:                                 7077 bytes
  uncompressed size:                               15348 bytes
  length of filename:                              3 characters
  length of extra field:                           0 bytes
  length of file comment:                          0 characters
  disk number on which file begins:                disk 1
  apparent file type:                              binary
  non-MSDOS external file attributes:              81B600 hex
  MS-DOS file attributes (20 hex):                 arc

  There is no file comment.

#
```

Appendix 1.2. Output of `strings -a -n 3 atd` command

Notes:
1. Strings command was executed with parameters:
`-a` – Scan the entire file, not just the data section;
`-n 3` - Locate & print any NUL-terminated sequence of at least 3 characters (default 4). I used this option, as it yielded few quite important strings that were skipped with default 4-character setting.
Running this program outputs all NUL-terminated strings that have 3 or more characters. This program behavior produces, together with meaningful strings some "noise" – sequences of characters that formally meet criteria of search, but in fact are just bytes of binary code that match the search pattern.
I filtered out these "noise" strings, as they do not add any information for forensic research. Some strings were repeated 8 times in a row – these were also filtered out.
2. Printout of command was re-formatted in 2 columns – for space considerations.

```
# strings -a -n 3 atd
```

```
ELF
/lib/ld-linux.so.1
libc.so.5
longjmp
strcpy
ioctl
popen
shmctl
geteuid
_DYNAMIC
getprotobynumber
errno
__strtol_internal
usleep
semget
getpid
fgets
shmat
_IO_stderr_
perror
getuid
semctl
optarg
socket
__environ
bzero
_init
alarm
__libc_init
environ
fprintf
kill
inet_addr
chdir
shmdt
setsockopt
__fpu_control
shmget
wait
umask
signal
read
strncmp
sendto
bcopy
fork
strdup
getopt
inet_ntoa
getppid
time
gethostbyname
_fini
sprintf
difftime
atexit
_GLOBAL_OFFSET_TABLE_
semop
exit
__setfpucw
open
setsid
close
_errno
_etext
_edata
__bss_start
_end
lokid: Client database full
DEBUG: stat_client nono
2.0
lokid version:          %s
remote interface:       %s
active transport:       %s
XOR
active cryptography:    %s
server uptime:
%.02f minutes
client ID:              %d
packets written:        %ld
bytes written:          %ld
requests:               %d
N@[fatal] cannot catch
SIGALRM
lokid: inactive client <%d>
expired from list [%d]
@[fatal] shared mem segment
request error
[fatal] semaphore
allocation error
[fatal] could not lock
memory
[fatal] could not unlock
memory
```

```
[fatal] shared mem segment
detach error
[fatal] cannot destroy
shmid
[fatal] cannot destroy
semaphore
[fatal] name lookup failed
[fatal] cannot catch
SIGALRM
[fatal] cannot catch
SIGCHLD
[fatal] Cannot go daemon
[fatal] Cannot create
session
/dev/tty
[fatal] cannot detach from
controlling terminal
/tmp
[fatal] invalid user
identification value
v:p:
Unknown transport
lokid -p (i|u) [ -v (0|1) ]
[fatal] socket allocation
error
[fatal] cannot catch
SIGUSR1
Cannot set IP_HDRINCL
socket option
[fatal] cannot register
with atexit(2)
LOKI2   route [(c) 1997
guild corporation
worldwide]
[fatal] cannot catch
SIGALRM
[fatal] cannot catch
SIGCHLD
[SUPER fatal] control
should NEVER fall here
[fatal] forking error
lokid: server is currently
at capacity.  Try again
later
lokid: Cannot add key
lokid: popen
[non fatal] truncated write
/quit all
lokid: client <%d>
requested an all kill
        sending L_QUIT:
<%d> %s
lokid: clean exit (killed
at client request)
[fatal] could not signal
process group
/quit
lokid: cannot locate client
entry in database
lokid: client <%d> freed
from list [%d]
/stat
/swapt
[fatal] could not signal
parent
lokid: unsupported or
unknown command string
lokid: client <%d>
requested a protocol swap
        sending protocol
update: <%d> %s [%d]
lokid: transport protocol
changed to %s
GCC: (GNU) 2.7.2.1
01.01
.symtab
.strtab
.shstrtab
.interp
.hash
.dynsym
.dynstr
.rel.bss
.rel.plt
.init
.plt
.text
.fini
.rodata
.data
.ctors
.dtors
.got
.dynamic
```

```
.bss                                    .note
.comment
#
```

Appendix 1.3. Results of running command readelf –a atd

Note: Command readelf shows detailed information about external references, symbols, libraries, system calls and other information about executable binary.
Parameter –a – for "all" categories of output.
I numbered lines for easier reference.
Interesting lines: 55, 70

```
1: # readelf -a atd
2: ELF Header:
3:   Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
4:   Class:                        ELF32
5:   Data:                         2's complement, little endian
6:   Version:                      1 (current)
7:   OS/ABI:                       UNIX - System V
8:   ABI Version:                  0
9:   Type:                         EXEC (Executable file)
10:  Machine:                      Intel 80386
11:  Version:                      0x1
12:  Entry point address:          0x8048db0
13:  Start of program headers:     52 (bytes into file)
14:  Start of section headers:     14508 (bytes into file)
15:  Flags:                        0x0
16:  Size of this header:          52 (bytes)
17:  Size of program headers:      32 (bytes)
18:  Number of program headers:    5
19:  Size of section headers:      40 (bytes)
20:  Number of section headers:    21
21:  Section header string table index: 20
22:
23: Section Headers:
24:  [Nr] Name              Type            Addr     Off    Size   ES Flg Lk Inf Al
25:  [ 0]                   NULL            00000000 000000 000000 00      0   0  0
26:  [ 1] .interp           PROGBITS        080480d4 0000d4 000013 00   A  0   0  1
27:  [ 2] .hash             HASH            080480e8 0000e8 0001a4 04   A  3   0  4
28:  [ 3] .dynsym           DYNSYM          0804828c 00028c 000420 10   A  4   1  4
29:  [ 4] .dynstr           STRTAB          080486ac 0006ac 000210 00   A  0   0  1
30:  [ 5] .rel.bss          REL             080488bc 0008bc 000020 08   A  3  11  4
31:  [ 6] .rel.plt          REL             080488dc 0008dc 000190 08   A  3   8  4
32:  [ 7] .init             PROGBITS        08048a70 000a70 000008 00  AX  0   0 16
33:  [ 8] .plt              PROGBITS        08048a78 000a78 000330 04  AX  0   0  4
34:  [ 9] .text             PROGBITS        08048db0 000db0 001b28 00  AX  0   0 16
35:  [10] .fini             PROGBITS        0804a8e0 0028e0 000008 00  AX  0   0 16
36:  [11] .rodata           PROGBITS        0804a8e8 0028e8 000c3c 00   A  0   0  4
37:  [12] .data             PROGBITS        0804c528 003528 000038 00  WA  0   0  4
```

```
38:  [13] .ctors          PROGBITS        0804c560 003560 000008 00  WA  0   0  4
39:  [14] .dtors          PROGBITS        0804c568 003568 000008 00  WA  0   0  4
40:  [15] .got            PROGBITS        0804c570 003570 0000d4 04  WA  0   0  4
41:  [16] .dynamic        DYNAMIC         0804c644 003644 000088 08  WA  4   0  4
42:  [17] .bss            NOBITS          0804c6cc 0036cc 00012c 00  WA  0   0  8
43:  [18] .comment        PROGBITS        00000000 0036cc 0000a0 00      0   0  1
44:  [19] .note           NOTE            000000a0 00376c 0000a0 00      0   0  1
45:  [20] .shstrtab       STRTAB          00000000 00380c 0000a0 00      0   0  1
46: Key to Flags:
47:  W (write), A (alloc), X (execute), M (merge), S (strings)
48:  I (info), L (link order), G (group), x (unknown)
49:  O (extra OS processing required) o (OS specific), p (processor specific)
50:
51: Program Headers:
52:  Type           Offset   VirtAddr   PhysAddr   FileSiz MemSiz  Flg Align
53:  PHDR           0x000034 0x08048034 0x08048034 0x000a0 0x000a0 R E 0x4
54:  INTERP         0x0000d4 0x080480d4 0x080480d4 0x00013 0x00013 R   0x1
55:      [Requesting program interpreter: /lib/ld-linux.so.1]
56:  LOAD           0x000000 0x08048000 0x08048000 0x03524 0x03524 R E
0x1000
57:  LOAD           0x003528 0x0804c528 0x0804c528 0x001a4 0x002d0 RW
0x1000
58:  DYNAMIC        0x003644 0x0804c644 0x0804c644 0x00088 0x00088 RW  0x4
59:
60:  Section to Segment mapping:
61:  Segment Sections...
62:   00
63:   01     .interp
64:   02     .interp .hash .dynsym .dynstr .rel.bss .rel.plt .init .plt .text .fini .rodata
65:   03     .data .ctors .dtors .got .dynamic .bss
66:   04     .dynamic
67:
68: Dynamic segment at offset 0x3644 contains 17 entries:
69:  Tag        Type                         Name/Value
70:  0x00000001 (NEEDED)                     Shared library: [libc.so.5]
71:  0x0000000c (INIT)                       0x8048a70
72:  0x0000000d (FINI)                       0x804a8e0
73:  0x00000004 (HASH)                       0x80480e8
74:  0x00000005 (STRTAB)                     0x80486ac
75:  0x00000006 (SYMTAB)                     0x804828c
76:  0x0000000a (STRSZ)                      528 (bytes)
77:  0x0000000b (SYMENT)                     16 (bytes)
78:  0x00000015 (DEBUG)                      0x0
79:  0x00000003 (PLTGOT)                     0x804c570
80:  0x00000002 (PLTRELSZ)                   400 (bytes)
81:  0x00000014 (PLTREL)                     REL
```

```
82:  0x00000017 (JMPREL)              0x80488dc
83:  0x00000011 (REL)                 0x80488bc
84:  0x00000012 (RELSZ)               32 (bytes)
85:  0x00000013 (RELENT)              8 (bytes)
86:  0x00000000 (NULL)                0x0
87:
88: Relocation section '.rel.bss' at offset 0x8bc contains 4 entries:
89:  Offset     Info    Type            Sym.Value  Sym. Name
90: 0804c6d8  00001005 R_386_COPY        0804c6d8   _IO_stderr_
91: 0804c72c  00001405 R_386_COPY        0804c72c   optarg
92: 0804c730  00002205 R_386_COPY        0804c730   __fpu_control
93: 0804c6d0  00003d05 R_386_COPY        0804c6d0   _errno
94:
95: Relocation section '.rel.plt' at offset 0x8dc contains 50 entries:
96:  Offset     Info    Type            Sym.Value  Sym. Name
97: 0804c57c  00000107 R_386_JUMP_SLOT   08048a88   longjmp
98: 0804c580  00000207 R_386_JUMP_SLOT   08048a98   strcpy
99: 0804c584  00000307 R_386_JUMP_SLOT   08048aa8   ioctl
100: 0804c588  00000407 R_386_JUMP_SLOT   08048ab8   popen
101: 0804c58c  00000507 R_386_JUMP_SLOT   08048ac8   shmctl
102: 0804c590  00000607 R_386_JUMP_SLOT   08048ad8   geteuid
103: 0804c594  00000807 R_386_JUMP_SLOT   08048ae8   getprotobynumber
104: 0804c598  00000a07 R_386_JUMP_SLOT   08048af8   __strtol_internal
105: 0804c59c  00000b07 R_386_JUMP_SLOT   08048b08   usleep
106: 0804c5a0  00000c07 R_386_JUMP_SLOT   08048b18   semget
107: 0804c5a4  00000d07 R_386_JUMP_SLOT   08048b28   getpid
108: 0804c5a8  00000e07 R_386_JUMP_SLOT   08048b38   fgets
109: 0804c5ac  00000f07 R_386_JUMP_SLOT   08048b48   shmat
110: 0804c5b0  00001107 R_386_JUMP_SLOT   08048b58   perror
111: 0804c5b4  00001207 R_386_JUMP_SLOT   08048b68   getuid
112: 0804c5b8  00001307 R_386_JUMP_SLOT   08048b78   semctl
113: 0804c5bc  00001507 R_386_JUMP_SLOT   08048b88   socket
114: 0804c5c0  00001707 R_386_JUMP_SLOT   08048b98   bzero
115: 0804c5c4  00001907 R_386_JUMP_SLOT   08048ba8   alarm
116: 0804c5c8  00001a07 R_386_JUMP_SLOT   08048bb8   __libc_init
117: 0804c5cc  00001c07 R_386_JUMP_SLOT   08048bc8   fprintf
118: 0804c5d0  00001d07 R_386_JUMP_SLOT   08048bd8   kill
119: 0804c5d4  00001e07 R_386_JUMP_SLOT   08048be8   inet_addr
120: 0804c5d8  00001f07 R_386_JUMP_SLOT   08048bf8   chdir
121: 0804c5dc  00002007 R_386_JUMP_SLOT   08048c08   shmdt
122: 0804c5e0  00002107 R_386_JUMP_SLOT   08048c18   setsockopt
123: 0804c5e4  00002307 R_386_JUMP_SLOT   08048c28   shmget
124: 0804c5e8  00002407 R_386_JUMP_SLOT   08048c38   wait
125: 0804c5ec  00002507 R_386_JUMP_SLOT   08048c48   umask
126: 0804c5f0  00002607 R_386_JUMP_SLOT   08048c58   signal
127: 0804c5f4  00002707 R_386_JUMP_SLOT   08048c68   read
```

```
128: 0804c5f8  00002807 R_386_JUMP_SLOT   08048c78   strncmp
129: 0804c5fc  00002907 R_386_JUMP_SLOT   08048c88   sendto
130: 0804c600  00002a07 R_386_JUMP_SLOT   08048c98   bcopy
131: 0804c604  00002b07 R_386_JUMP_SLOT   08048ca8   fork
132: 0804c608  00002c07 R_386_JUMP_SLOT   08048cb8   strdup
133: 0804c60c  00002d07 R_386_JUMP_SLOT   08048cc8   getopt
134: 0804c610  00002e07 R_386_JUMP_SLOT   08048cd8   inet_ntoa
135: 0804c614  00002f07 R_386_JUMP_SLOT   08048ce8   getppid
136: 0804c618  00003007 R_386_JUMP_SLOT   08048cf8   time
137: 0804c61c  00003107 R_386_JUMP_SLOT   08048d08   gethostbyname
138: 0804c620  00003307 R_386_JUMP_SLOT   08048d18   sprintf
139: 0804c624  00003407 R_386_JUMP_SLOT   08048d28   difftime
140: 0804c628  00003507 R_386_JUMP_SLOT   08048d38   atexit
141: 0804c62c  00003707 R_386_JUMP_SLOT   08048d48   semop
142: 0804c630  00003807 R_386_JUMP_SLOT   08048d58   exit
143: 0804c634  00003907 R_386_JUMP_SLOT   08048d68   __setfpucw
144: 0804c638  00003a07 R_386_JUMP_SLOT   08048d78   open
145: 0804c63c  00003b07 R_386_JUMP_SLOT   08048d88   setsid
146: 0804c640  00003c07 R_386_JUMP_SLOT   08048d98   close
147:
148: There are no unwind sections in this file.
149:
150: Symbol table '.dynsym' contains 66 entries:
151:    Num:    Value  Size Type    Bind   Vis      Ndx Name
152:      0: 00000000     0 NOTYPE  LOCAL  DEFAULT  UND
153:      1: 08048a88     0 FUNC    GLOBAL DEFAULT  UND longjmp
154:      2: 08048a98    30 FUNC    GLOBAL DEFAULT  UND strcpy
155:      3: 08048aa8     0 FUNC    WEAK   DEFAULT  UND ioctl
156:      4: 08048ab8     0 FUNC    WEAK   DEFAULT  UND popen
157:      5: 08048ac8    42 FUNC    GLOBAL DEFAULT  UND shmctl
158:      6: 08048ad8     0 FUNC    WEAK   DEFAULT  UND geteuid
159:      7: 0804c644     0 OBJECT  GLOBAL DEFAULT  ABS _DYNAMIC
160:      8: 08048ae8   292 FUNC    GLOBAL DEFAULT  UND getprotobynumber
161:      9: 0804c6d0     4 NOTYPE  WEAK   DEFAULT   17 errno
162:     10: 08048af8  1132 FUNC    GLOBAL DEFAULT  UND __strtol_internal
163:     11: 08048b08    99 FUNC    GLOBAL DEFAULT  UND usleep
164:     12: 08048b18    42 FUNC    GLOBAL DEFAULT  UND semget
165:     13: 08048b28     0 FUNC    WEAK   DEFAULT  UND getpid
166:     14: 08048b38     0 FUNC    WEAK   DEFAULT  UND fgets
167:     15: 08048b48    59 FUNC    GLOBAL DEFAULT  UND shmat
168:     16: 0804c6d8    84 OBJECT  GLOBAL DEFAULT   17 _IO_stderr_
169:     17: 08048b58     0 FUNC    WEAK   DEFAULT  UND perror
170:     18: 08048b68     0 FUNC    WEAK   DEFAULT  UND getuid
171:     19: 08048b78    47 FUNC    GLOBAL DEFAULT  UND semctl
172:     20: 0804c72c     4 OBJECT  GLOBAL DEFAULT   17 optarg
173:     21: 08048b88    94 FUNC    WEAK   DEFAULT  UND socket
```

```
174:    22: 0804c528     4 OBJECT  GLOBAL DEFAULT   12 __environ
175:    23: 08048b98    54 FUNC    GLOBAL DEFAULT  UND bzero
176:    24: 08048a70     0 FUNC    GLOBAL DEFAULT    7 _init
177:    25: 08048ba8     0 FUNC    WEAK   DEFAULT  UND alarm
178:    26: 08048bb8    70 FUNC    GLOBAL DEFAULT  UND __libc_init
179:    27: 0804c528     4 NOTYPE  WEAK   DEFAULT   12 environ
180:    28: 08048bc8     0 FUNC    WEAK   DEFAULT  UND fprintf
181:    29: 08048bd8     0 FUNC    WEAK   DEFAULT  UND kill
182:    30: 08048be8    57 FUNC    GLOBAL DEFAULT  UND inet_addr
183:    31: 08048bf8     0 FUNC    WEAK   DEFAULT  UND chdir
184:    32: 08048c08    36 FUNC    GLOBAL DEFAULT  UND shmdt
185:    33: 08048c18   111 FUNC    WEAK   DEFAULT  UND setsockopt
186:    34: 0804c730     2 OBJECT  GLOBAL DEFAULT   17 __fpu_control
187:    35: 08048c28    42 FUNC    GLOBAL DEFAULT  UND shmget
188:    36: 08048c38     0 FUNC    WEAK   DEFAULT  UND wait
189:    37: 08048c48     0 FUNC    WEAK   DEFAULT  UND umask
190:    38: 08048c58    84 FUNC    GLOBAL DEFAULT  UND signal
191:    39: 08048c68     0 FUNC    WEAK   DEFAULT  UND read
192:    40: 08048c78    38 FUNC    GLOBAL DEFAULT  UND strncmp
193:    41: 08048c88   124 FUNC    WEAK   DEFAULT  UND sendto
194:    42: 08048c98   146 FUNC    GLOBAL DEFAULT  UND bcopy
195:    43: 08048ca8     0 FUNC    WEAK   DEFAULT  UND fork
196:    44: 08048cb8    79 FUNC    GLOBAL DEFAULT  UND strdup
197:    45: 08048cc8    44 FUNC    GLOBAL DEFAULT  UND getopt
198:    46: 08048cd8    67 FUNC    GLOBAL DEFAULT  UND inet_ntoa
199:    47: 08048ce8     0 FUNC    WEAK   DEFAULT  UND getppid
200:    48: 08048cf8     0 FUNC    WEAK   DEFAULT  UND time
201:    49: 08048d08   292 FUNC    GLOBAL DEFAULT  UND gethostbyname
202:    50: 0804a8e0     0 FUNC    GLOBAL DEFAULT   10 _fini
203:    51: 08048d18    38 FUNC    WEAK   DEFAULT  UND sprintf
204:    52: 08048d28    16 FUNC    GLOBAL DEFAULT  UND difftime
205:    53: 08048d38    52 FUNC    GLOBAL DEFAULT  UND atexit
206:    54: 0804c570     0 OBJECT  GLOBAL DEFAULT  ABS
_GLOBAL_OFFSET_TABLE_
207:    55: 08048d48    42 FUNC    GLOBAL DEFAULT  UND semop
208:    56: 08048d58   128 FUNC    GLOBAL DEFAULT  UND exit
209:    57: 08048d68    62 FUNC    GLOBAL DEFAULT  UND __setfpucw
210:    58: 08048d78     0 FUNC    WEAK   DEFAULT  UND open
211:    59: 08048d88     0 FUNC    WEAK   DEFAULT  UND setsid
212:    60: 08048d98     0 FUNC    WEAK   DEFAULT  UND close
213:    61: 0804c6d0     4 OBJECT  GLOBAL DEFAULT   17 _errno
214:    62: 0804a8d8     0 OBJECT  GLOBAL DEFAULT  ABS _etext
215:    63: 0804c6cc     0 OBJECT  GLOBAL DEFAULT  ABS _edata
216:    64: 0804c6cc     0 OBJECT  GLOBAL DEFAULT  ABS __bss_start
217:    65: 0804c7f8     0 OBJECT  GLOBAL DEFAULT  ABS _end
218:
```

```
Histogram for bucket list length (total of 37 buckets):
 Length  Number     % of total  Coverage
      0  9          ( 24.3%)
      1  8          ( 21.6%)     12.3%
      2  10         ( 27.0%)     43.1%
      3  4          ( 10.8%)     61.5%
      4  5          ( 13.5%)     92.3%
      5  1          (  2.7%)    100.0%

No version information found in this file.
#
```

Appendix 1.4. Diagnostic messages during initial compilation of LOKI2 program

```
# make linux
make[1]: Entering directory `/loki/L2'
gcc -Wall -O6 -finline-functions -funroll-all-loops -DLINUX
-DWEAK_CRYPTO -DPOPEN -DSEND_PAUSE=100 -Dx86_FAST_CHECK
-c surplus.c -o surplus.o
In file included from loki.h:36,
                 from surplus.c:10:
/usr/include/linux/icmp.h:67: parse error before "__u8"
/usr/include/linux/icmp.h:69: parse error before "checksum"
/usr/include/linux/icmp.h:72: parse error before "__u16"
/usr/include/linux/icmp.h:75: parse error before "gateway"
/usr/include/linux/icmp.h:77: parse error before "__u16"
/usr/include/linux/icmp.h:80: parse error before '}' token
/usr/include/linux/icmp.h:81: parse error before '}' token
/usr/include/linux/icmp.h:90: parse error before "__u32"
In file included from /usr/include/linux/signal.h:4,
                 from loki.h:38,
                 from surplus.c:10:
/usr/include/asm/signal.h:102: redefinition of `struct
sigaction'
/usr/include/asm/signal.h:116: redefinition of `struct
sigaltstack'
In file included from /usr/include/linux/signal.h:5,
                 from loki.h:38,
                 from surplus.c:10:
/usr/include/asm/siginfo.h:8: redefinition of `union
sigval'
/usr/include/asm/siginfo.h:16: redefinition of `struct
siginfo'
/usr/include/asm/siginfo.h:199: redefinition of `struct
sigevent'
make[1]: *** [surplus.o] Error 1
make[1]: Leaving directory `/loki/L2'
make: *** [linux] Error 2
#
```

Appendix 1.5. Results of running program strace ./atd

Note: Lines are numbered for convenience of reference.

```
execve("./atd", ["./atd"], [/* 32 vars */]) = 0
old_mmap(NULL, 4096, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x40007000
mprotect(0x40000000, 21772, PROT_READ|PROT_WRITE|PROT_EXEC) =
0
mprotect(0x8048000, 13604, PROT_READ|PROT_WRITE|PROT_EXEC) = 0
stat("/etc/ld.so.cache", {st_mode=S_IFREG|0644, st_size=50656, ...}) = 0
open("/etc/ld.so.cache", O_RDONLY)      = 3
old_mmap(NULL, 50656, PROT_READ, MAP_SHARED, 3, 0) = 0x40008000
close(3)                                = 0
stat("/etc/ld.so.preload", 0xbffff990)  = -1 ENOENT (No such file or directory)
open("/usr/i486-linux-libc5/lib/libc.so.5", O_RDONLY) = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0\310\234"..., 4096) =
4096
old_mmap(NULL, 884736, PROT_NONE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x40015000
old_mmap(0x40015000, 647729, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED, 3, 0) = 0x40015000
old_mmap(0x400b4000, 22104, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED, 3, 0x9e000) = 0x400b4000
old_mmap(0x400ba000, 205560, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x400ba000
close(3)                                = 0
mprotect(0x40015000, 647729, PROT_READ|PROT_WRITE|PROT_EXEC)
= 0
munmap(0x40008000, 50656)               = 0
mprotect(0x8048000, 13604, PROT_READ|PROT_EXEC) = 0
mprotect(0x40015000, 647729, PROT_READ|PROT_EXEC) = 0
mprotect(0x40000000, 21772, PROT_READ|PROT_EXEC) = 0
personality(0 /* PER_??? */)            = 0
geteuid()                               = 0
getuid()                                = 0
getgid()                                = 0
getegid()                               = 0
geteuid()                               = 0
getuid()                                = 0
brk(0x804c820)                          = 0x804c820
brk(0x804d000)                          = 0x804d000
open("/usr/share/locale/locale.alias", O_RDONLY) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=2601, ...}) = 0
```

```
33: old_mmap(NULL, 4096, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x40008000
34: read(3, "# Locale name alias data base.\n#"..., 4096) = 2601
35: brk(0x804e000)                      = 0x804e000
36: read(3, "", 4096)                   = 0
37: close(3)                            = 0
38: munmap(0x40008000, 4096)            = 0
39: open("/usr/share/i18n/locale.alias", O_RDONLY) = -1 ENOENT (No such file
or directory)
40: open("/usr/share/locale/en_US.UTF-8/LC_MESSAGES", O_RDONLY) = -1
ENOENT (No such file or directory)
41: open("/usr/share/locale/en_US.utf8/LC_MESSAGES", O_RDONLY) = -1
ENOENT (No such file or directory)
42: open("/usr/share/locale/en_US/LC_MESSAGES", O_RDONLY) = 3
43: fstat(3, {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
44: close(3)                            = 0
45: open("/usr/share/locale/en_US/LC_MESSAGES/SYS_LC_MESSAGES",
O_RDONLY) = -1 ENOENT (No such file or directory)
46: open("/usr/share/locale/en.UTF-8/LC_MESSAGES", O_RDONLY) = -1
ENOENT (No such file or directory)
47: open("/usr/share/locale/en.utf8/LC_MESSAGES", O_RDONLY) = -1
ENOENT (No such file or directory)
48: open("/usr/share/locale/en/LC_MESSAGES", O_RDONLY) = 3
49: fstat(3, {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
50: close(3)                            = 0
51: open("/usr/share/locale/en/LC_MESSAGES/SYS_LC_MESSAGES",
O_RDONLY) = -1 ENOENT (No such file or directory)
52: stat("/etc/locale/C/libc.cat", 0xbffff4b0) = -1 ENOENT (No such file or
directory)
53: stat("/usr/share/locale/C/libc.cat", 0xbffff4b0) = -1 ENOENT (No such file or
directory)
54: stat("/usr/share/locale/libc/C", 0xbffff4b0) = -1 ENOENT (No such file or
directory)
55: stat("/usr/share/locale/C/libc.cat", 0xbffff4b0) = -1 ENOENT (No such file or
directory)
56: stat("/usr/local/share/locale/C/libc.cat", 0xbffff4b0) = -1 ENOENT (No such file
or directory)
57: socket(PF_INET, SOCK_RAW, IPPROTO_ICMP) = 3
58: sigaction(SIGUSR1, {0x804a6b0, [],
SA_INTERRUPT|SA_NOMASK|SA_ONESHOT}, {SIG_DFL}, 0x4004e238) = 0
59: socket(PF_INET, SOCK_RAW, IPPROTO_RAW)  = 4
60: setsockopt(4, SOL_IP, IP_HDRINCL, [1], 4) = 0
61: getpid()                            = 1264
62: getpid()                            = 1264
63: shmget(1506, 240, IPC_CREAT|0)          = 2031639
64: semget(1688, 1, IPC_CREAT|0x180|0600)   = 98307
```

```
65: shmat(2031639, 0, 0)                = 0x40008000
66: write(2, "\nLOKI2\troute [(c) 1997 guild cor"..., 52) = 52
67: time([1043900901])                  = 1043900901
68: close(0)                            = 0
69: sigaction(SIGTTOU, {SIG_IGN}, {SIG_DFL}, 0x4004e238) = 0
70: sigaction(SIGTTIN, {SIG_IGN}, {SIG_DFL}, 0x4004e238) = 0
71: sigaction(SIGTSTP, {SIG_IGN}, {SIG_DFL}, 0x4004e238) = 0
72: fork()                              = 1265
73: close(4)                            = 0
74: close(3)                            = 0
75: semop(98307, 0xbffff934, 2)         = 0
76: shmdt(0x40008000)                   = 0
77: semop(98307, 0xbffff934, 1)         = 0
78: _exit(0)                            = ?
```

Appendix 1.6. Output of `strings -a -n 3 lokid` command

Notes:
1. Strings command was executed with parameters:
`-a` – Scan the entire file, not just the data section;
`-n 3` - Locate & print any NUL-terminated sequence of at least 3 characters (default 4). I used this option, as it yielded few quite important strings that were skipped with default 4-character setting.
Running this program outputs all NUL-terminated strings that have 3 or more characters. This program behavior produces, together with meaningful strings some "noise" – sequences of characters that formally meet criteria of search, but in fact are just bytes of binary code that match the search pattern.
I filtered out these "noise" strings, as they do not add any information for forensic research. Some strings were repeated 8 times in a row – these were also filtered out.
2. Printout of command was re-formatted in 2 columns – for space considerations.

```
# strings -a -n 3 lokid
```

ELF
/lib/ld-linux.so.2
GNU
libc.so.6
longjmp
strcpy
ioctl
geteuid
getprotobynumber
errno
__strtol_internal
usleep
semget
getpid
fgets
shmat
perror
getuid
optarg
socket
alarm
popen
fprintf
kill
inet_addr
chdir
shmdt
setsockopt
shmget
wait
umask
signal
read
sendto
__strdup
bcopy
fork
getopt
inet_ntoa
getppid
gethostbyname
sprintf
difftime
stderr
shmctl
fwrite
semctl
semop
_IO_stdin_used
__libc_start_main

```
setsid
close
__cxa_atexit
__gmon_start__
GLIBC_2.2
GLIBC_2.1
GLIBC_2.1.3
GLIBC_2.0
PTRh@
QVh
WVS
f;<
Ff;<
[^_
WVS
[^_
WVS
;h"
[^_
WVS
[^_
WVS
[^_
WVS
[^_
WVS
WVS
[^_
tJPj
@t3j
Y[j
XZj
x Qj
h"T
@t4
ZYP
XZh
XZh$
VS1
ux9
WVS
pu@
j4j
Y_h
XZh$
PjTh
Pj7Sh
Sj7j
j7V
Qj8SV
j7V
Rj8SV
j7V
tZPj8SV
j7V
t%Pj8SV
 hS
j@h
5t'O
j@h
it$
WVS1
Sj7h=
ZYj
jTh
[^_
j@f
j@f
WVj7j
WVS
tbj
[^_
|)j
Pj7
Pj7
<1W
Pj7
<0W
Pj7
@t?
Wh
j.j
Rh
WV1
XZh
[^_
lokid: Client database full
2.0
lokid version:      %s
remote interface:   %s
active transport:   %s
XOR
```

```
active cryptography:%s
client ID:          %d
packets written:    %ld
bytes written:      %ld
requests:           %d
DEBUG: stat_client nono
[fatal] cannot catch SIGALRM
server uptime:              %.02f
minutes
lokid: inactive client <%d> expired
from list [%d]
[fatal] semaphore allocation error
[fatal] shared mem segment request
error
[fatal] could not unlock memory
[fatal] cannot destroy semaphore
[fatal] shared mem segment detach
error
[fatal] could not lock memory
[fatal] cannot destroy shmid
/dev/tty
/tmp
[fatal] Cannot create session
[fatal] Cannot go daemon
[fatal] name lookup failed
[fatal] cannot catch SIGCHLD
[fatal] cannot detach from controlling
terminal
[fatal] invalid user identification value
LOKI2 route [(c) 1997 guild
corporation worldwide]
[SUPER fatal] control should NEVER
fall here
lokid: server is currently at capacity.
Try again later
[fatal] cannot register with atexit(2)
Cannot set IP_HDRINCL socket
option
[fatal] socket allocation error
lokid: unsupported or unknown
command string
[fatal] could not signal parent
lokid: client <%d> freed from list [%d]
lokid: cannot locate client entry in
database
[fatal] could not signal process group
lokid: clean exit (killed at client
request)
lokid: client <%d> requested an all
kill
lokid: transport protocol changed to
%s
        sending protocol update:
<%d> %s [%d]
lokid: client <%d> requested a
protocol swap
v:p:
lokid -p (i|u) [ -v (0|1) ]
Unknown transport
lokid: popen
lokid: Cannot add key
[fatal] forking error
[fatal] cannot catch SIGUSR1
[non fatal] truncated write
/quit all
/quit
/stat
/swapt
        sending L_QUIT: <%d> %s
GCC: (GNU) 3.2 20020903 (Red Hat
Linux 8.0 3.2-7)
GCC: (GNU) 3.2 20020903 (Red Hat
Linux 8.0 3.2-7)
GCC: (GNU) 3.2 20020903 (Red Hat
Linux 8.0 3.2-7)
GCC: (GNU) 3.2 20020903 (Red Hat
Linux 8.0 3.2-7)
GCC: (GNU) 3.2 20020903 (Red Hat
Linux 8.0 3.2-7)
GCC: (GNU) 3.2 20020903 (Red Hat
Linux 8.0 3.2-7)
GCC: (GNU) 3.2 20020903 (Red Hat
Linux 8.0 3.2-7)
GCC: (GNU) 3.2 20020903 (Red Hat
Linux 8.0 3.2-7)
GCC: (GNU) 3.2 20020903 (Red Hat
Linux 8.0 3.2-7)
GCC: (GNU) 3.2 20020903 (Red Hat
Linux 8.0 3.2-7)
GCC: (GNU) 3.2 20020903 (Red Hat
Linux 8.0 3.2-7)
.shstrtab
```

.interp
.note.ABI-tag
.hash
.dynsym
.dynstr
.gnu.version
.gnu.version_r
.rel.dyn
.rel.plt
.init
.text
.fini
.rodata
.data
.eh_frame
.dynamic
.ctors
.dtors
.jcr
.got
.bss
.comment

Appendix 1.7. Results of running program strace ./lokid

```
[root@lebenstern sans2]# strace ./lokid
execve("./lokid", ["./lokid"], [/* 21 vars */]) = 0
uname({sys="Linux", node="lebenstern.cch-lis.com", ...}) = 0
brk(0)                                  = 0x804c97c
open("/etc/ld.so.preload", O_RDONLY)    = -1 ENOENT (No such file or
directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=42118, ...}) = 0
old_mmap(NULL, 42118, PROT_READ, MAP_PRIVATE, 3, 0) = 0x40013000
close(3)                                = 0
open("/lib/i686/libc.so.6", O_RDONLY)   = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0\220Y\1"..., 1024) = 1024
fstat64(3, {st_mode=S_IFREG|0755, st_size=1395734, ...}) = 0
old_mmap(0x42000000, 1239844, PROT_READ|PROT_EXEC, MAP_PRIVATE,
3, 0) = 0x42000000
mprotect(0x42126000, 35620, PROT_NONE)  = 0
old_mmap(0x42126000, 20480, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED, 3, 0x126000) = 0x42126000
old_mmap(0x4212b000, 15140, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x4212b000
close(3)                                = 0
old_mmap(NULL, 4096, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x4001e000
munmap(0x40013000, 42118)               = 0
geteuid32()                             = 0
getuid32()                              = 0
socket(PF_INET, SOCK_RAW, IPPROTO_ICMP) = 3
rt_sigaction(SIGUSR1, {0x804aa94, [USR1], SA_RESTORER|SA_RESTART,
0x42028c48}, {SIG_DFL}, 8) = 0
socket(PF_INET, SOCK_RAW, IPPROTO_RAW)  = 4
setsockopt(4, SOL_IP, IP_HDRINCL, [1], 4) = 0
getpid()                                = 17082
getpid()                                = 17082
shmget(17324, 240, IPC_CREAT|0)         = 786442
semget(17506, 1, IPC_CREAT|0x180|0600)  = 262152
shmat(786442, 0, 0)                     = 0x40013000
write(2, "\nLOKI2\troute [(c) 1997 guild cor"..., 52
LOKI2   route [(c) 1997 guild corporation worldwide]
) = 52
time([1047525391])                      = 1047525391
close(0)                                = 0
rt_sigaction(SIGTTOU, {SIG_IGN}, {SIG_DFL}, 8) = 0
rt_sigaction(SIGTTIN, {SIG_IGN}, {SIG_DFL}, 8) = 0
```

```
rt_sigaction(SIGTSTP, {SIG_IGN}, {SIG_DFL}, 8) = 0
fork()                          = 17083
close(4)                        = 0
close(3)                        = 0
semop(262152, 0xbffffa00, 2)        = 0
shmdt(0x40013000)               = 0
semop(262152, 0xbffff9f0, 1)        = 0
_exit(0)                        = ?
[root@lebenstern sans2]#
```

Appendix 2.1. – Sample report produced by SecReport tool

Security Report AK2

Hostname: AK2
Date and time report, start: 2003-03-10, 21:05:09, (GMT-05:00)
Date and time report, finish: 2003-03-10, 21:05:42, (GMT-05:00)
Operating System: Microsoft Windows 2000 Server 5.0.2195
Service Pack: 3.0
Server Domain: WGZEIT
Server Role: Standalone Server
IE Version: 6.0.2800.1106
Java VM Version: 5.0.3805.0
Media Player Version: 6.4.9.1121
WSH Version: 5.6.0.6626

Logged on user: AK2\LocalAdmin

Network Configuration

NIC Brand and Model: 3Com EtherLink PCI
IP Address: 192.168.1.20 **Subnet Mask:** 255.255.255.0
Gateway: 192.168.1.1
DNS Server: 192.168.1.20
WINS Server: 192.168.1.20
WINS Server: 192.168.1.40
MAC Address: 00:50:04:21:96:46

Audit Policy

Policy	Security setting
Account Logon	Success and Failure
Account Management	Success and Failure
Directory Service Access	No
Logon	Success and Failure
Object Access	No
Policy Change	Success and Failure
Privilege Use	No
Process Tracking	No
System	Success and Failure

Event Log configuration

Log Name	Max Size (KB)	Overwrite Old Events	Filename
Application	512	Overwrite events older than 7 days	C:\WINNT\system32\config\AppEvent.Evt
Security	512	Overwrite events older than 7 days	C:\WINNT\System32\config\SecEvent.Evt
System	512	Overwrite events older than 7 days	C:\WINNT\system32\config\SysEvent.Evt

Applications

Number of applications: 24

Adobe Acrobat 5.0 5.0
CMD Prompt Here PowerToy
IE XML/XSL Viewer Tools
IIS UrlScan Tool 2.5 (Uninstall)
LiveUpdate 1.7 (Symantec Corporation)
Macromedia Dreamweaver 4 4.0
Macromedia Extension Manager 1.2
MetaEdit 2.2 (x86)
Microsoft Internet Explorer 6 SP1
Microsoft Office XP Professional with FrontPage 10.0.4330.0
Microsoft SQL Server 2000 8.00.534
Microsoft Windows 2000 Server Resource Kit: Supplement 1 5.0.2092.1
Microsoft XML Parser and SDK 4.10.9404.0
Nero - Burning Rom (Web installer)
Paint Shop Pro 7 7.0.2.0000
Run shell extension
Symantec AntiVirus Client 8.0.0.374
TextPad
Tweak UI
Van Dyke Technologies CRT 3.4 3.4
WebFldrs 9.00.3907
Windows Commander (Remove or Repair)
Windows Script V5.6 Documentation

WinZip 8.1 (4331)

Hotfixes

Number of hotfixes: 2
MS02-066 328970
MS02-068 324929

IIS Configuration

Anonymous username: IUSR_AK2	**Password:** rrWIBgqHOA@6Er
WAM username: IWAM_AK2	**Password:** v5GgSKsSWx)9Sd

IIS Log File Directory: %WinDir%\System32\LogFiles
URLScan Version: 6.0.3615.0

ISAPI Filters

Filter load order: UrlScan,sspifilt,Compression,md5filt,fpexedll.dll

Description	Path	Priority
Microsoft SSPI Encryption Filter, v1.0	C:\WINNT\System32\inetsrv\sspifilt.dll	High
HTTP 1.1 Compression filter version, v1.0	C:\WINNT\System32\inetsrv\compfilt.dll	High
Digest Authentication, version 1.0	C:\WINNT\System32\inetsrv\md5filt.dll	Low
Microsoft FrontPage Server Extensions	C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\40\bin\fpexedll.dll	Low
UrlScan ISAPI Filter	C:\WINNT\System32\inetsrv\urlscan\urlscan.dll	Low

DLL Application Mappings

Extension	Path	Flag	Verbs
.asp	C:\WINNT\System32\inetsrv\asp.dll	1	GET; HEAD; POST; TRACE;
.cer	C:\WINNT\System32\inetsrv\asp.dll	1	GET; HEAD; POST; TRACE;
.cdx	C:\WINNT\System32\inetsrv\asp.dll	1	GET; HEAD; POST; TRACE;
.asa	C:\WINNT\System32\inetsrv\asp.dll	1	GET; HEAD; POST; TRACE;
.htr	C:\WINNT\System32\inetsrv\404.dll	1	GET; POST;

.idc	C:\WINNT\System32\inetsrv\404.dll	1	OPTIONS; GET; HEAD; POST; PUT; DELETE; TRACE;
.shtm	C:\WINNT\System32\inetsrv\404.dll	1	GET; POST;
.shtml	C:\WINNT\System32\inetsrv\404.dll	1	GET; POST;
.stm	C:\WINNT\System32\inetsrv\404.dll	1	GET; POST;

Services

Total number of services: 71; Number of Running services: 28; Number of Automatic services: 23; Number of Manual services: 17

Service	Start Type	Status	Service full name	Account
Alerter	Disabled	Stopped	Alerter	LocalSystem
AppMgmt	Manual	Stopped	Application Management	LocalSystem
BITS	Manual	Stopped	Background Intelligent Transfer Service	LocalSystem
Browser	Disabled	Stopped	Computer Browser	LocalSystem
cisvc	Disabled	Stopped	Indexing Service	LocalSystem
ClipSrv	Disabled	Stopped	ClipBook	LocalSystem
DefWatch	Automatic	Running	DefWatch	LocalSystem
Dfs	Disabled	Stopped	Distributed File System	LocalSystem
Dhcp	Automatic	Running	DHCP Client	LocalSystem
dmadmin	Manual	Stopped	Logical Disk Manager Administrative Service	LocalSystem
dmserver	Automatic	Running	Logical Disk Manager	LocalSystem
Dnscache	Automatic	Running	DNS Client	LocalSystem
Eventlog	Automatic	Running	Event Log	LocalSystem
EventSystem	Manual	Running	COM+ Event System	LocalSystem
Fax	Disabled	Stopped	Fax Service	LocalSystem
IISADMIN	Manual	Running	IIS Admin Service	LocalSystem
IsmServ	Disabled	Stopped	Intersite Messaging	LocalSystem
kdc	Disabled	Stopped	Kerberos Key Distribution Center	LocalSystem
lanmanserver	Automatic	Running	Server	LocalSystem
lanmanworkstation	Automatic	Running	Workstation	LocalSystem
LicenseService	Disabled	Stopped	License Logging Service	LocalSystem
LmHosts	Disabled	Stopped	TCP/IP NetBIOS Helper	LocalSystem

			Service	
MDM	Disabled	Stopped	Machine Debug Manager	LocalSystem
Messenger	Disabled	Stopped	Messenger	LocalSystem
mnmsrvc	Disabled	Stopped	NetMeeting Remote Desktop Sharing	LocalSystem
MSDTC	Disabled	Stopped	Distributed Transaction Coordinator	LocalSystem
MSFTPSVC	Disabled	Stopped	FTP Publishing Service	LocalSystem
MSIServer	Manual	Stopped	Windows installer	LocalSystem
MSSEARCH	Automatic	Running	Microsoft Search	LocalSystem
MSSQLSERVER	Automatic	Running	MSSQLSERVER	.\SQLKonig
MSSQLServerADHelper	Manual	Stopped	MSSQLServerADHelper	LocalSystem
NetDDE	Disabled	Stopped	Network DDE	LocalSystem
NetDDEdsdm	Disabled	Stopped	Network DDE DSDM	LocalSystem
Netlogon	Manual	Stopped	Net Logon	LocalSystem
Netman	Manual	Running	Network Connections	LocalSystem
Norton AntiVirus Server	Automatic	Running	Symantec AntiVirus Client	LocalSystem
NtFrs	Disabled	Stopped	File Replication	LocalSystem
NtLmSsp	Manual	Running	NT LM Security Support Provider	LocalSystem
NtmsSvc	Automatic	Running	Removable Storage	LocalSystem
PlugPlay	Automatic	Running	Plug and Play	LocalSystem
PolicyAgent	Automatic	Running	IPSEC Policy Agent	LocalSystem
ProtectedStorage	Automatic	Running	Protected Storage	LocalSystem
RasAuto	Manual	Stopped	Remote Access Auto Connection Manager	LocalSystem
RasMan	Manual	Stopped	Remote Access Connection Manager	LocalSystem
RemoteAccess	Disabled	Stopped	Routing and Remote Access	LocalSystem
RemoteRegistry	Disabled	Stopped	Remote Registry Service	LocalSystem
RpcLocator	Disabled	Stopped	Remote Procedure Call (RPC) Locator	LocalSystem
RpcSs	Automatic	Running	Remote Procedure Call (RPC)	LocalSystem
RSVP	Manual	Stopped	QoS RSVP	LocalSystem
SamSs	Automatic	Running	Security Accounts Manager	LocalSystem
SCardDrv	Disabled	Stopped	Smart Card Helper	LocalSystem

SCardSvr	Disabled	Stopped	Smart Card	LocalSystem
Schedule	Automatic	Running	Task Scheduler	LocalSystem
seclogon	Automatic	Running	RunAs Service	LocalSystem
SENS	Automatic	Running	System Event Notification	LocalSystem
SharedAccess	Disabled	Stopped	Internet Connection Sharing	LocalSystem
Spooler	Automatic	Running	Print Spooler	LocalSystem
SQLSERVERAGENT	Manual	Stopped	SQLSERVERAGENT	.\SQLKonig
SysmonLog	Manual	Stopped	Performance Logs and Alerts	LocalSystem
TapiSrv	Disabled	Stopped	Telephony	LocalSystem
TermService	Disabled	Stopped	Terminal Services	LocalSystem
TlntSvr	Disabled	Stopped	Telnet	LocalSystem
TrkSvr	Disabled	Stopped	Distributed Link Tracking Server	LocalSystem
TrkWks	Disabled	Stopped	Distributed Link Tracking Client	LocalSystem
UPS	Disabled	Stopped	Uninterruptible Power Supply	LocalSystem
UtilMan	Disabled	Stopped	Utility Manager	LocalSystem
W32Time	Manual	Stopped	Windows Time	LocalSystem
W3SVC	Automatic	Running	World Wide Web Publishing Service	LocalSystem
WinMgmt	Automatic	Running	Windows Management Instrumentation	LocalSystem
Wmi	Manual	Running	Windows Management Instrumentation Driver Extensions	LocalSystem
wuauserv	Automatic	Running	Automatic Updates	LocalSystem

Ports open

Total number of open ports: 14; Number of open TCP ports: 8; Number of open UDP ports: 6

Port	Protocol	PID	Program short name	Program long name
80	TCP	752	inetinfo	C:\WINNT\System32\inetsrv\inetinfo.exe
135	TCP	416	svchost	C:\WINNT\system32\svchost.exe
139	TCP	8	System	

443	TCP	752	inetinfo	C:\WINNT\System32\inetsrv\inetinfo.exe
445	TCP	8	System	
1025	TCP	648	MSTask	C:\WINNT\system32\MSTask.exe
1028	TCP	752	inetinfo	C:\WINNT\System32\inetsrv\inetinfo.exe
1433	TCP	536	sqlservr	C:\PROGRA~1\MI6841~1\MSSQL\binn\sqlservr.exe
137	UDP	8	System	
138	UDP	8	System	
445	UDP	8	System	
500	UDP	244	lsass	C:\WINNT\system32\lsass.exe
1434	UDP	536	sqlservr	C:\PROGRA~1\MI6841~1\MSSQL\binn\sqlservr.exe
3456	UDP	752	inetinfo	C:\WINNT\System32\inetsrv\inetinfo.exe

Processes active

Total number of active processes: 26

Process Name	PID	Parent PID	Thread Count	Handle Count	Command Line
System Idle Process	0	0	1	0	
System	8	0	38	162	
smss.exe	160	8	6	33	C:\WINNT\System32\smss.exe
csrss.exe	184	160	11	405	
winlogon.exe	204	160	19	395	C:\WINNT\system32\winlogon.exe
services.exe	232	204	31	483	C:\WINNT\system32\services.exe
lsass.exe	244	204	17	254	C:\WINNT\system32\lsass.exe
svchost.exe	416	232	10	273	C:\WINNT\system32\svchost.exe
SPOOLSV.EXE	448	232	14	184	C:\WINNT\system32\spoolsv.exe
DefWatch.exe	476	232	4	45	C:\Program Files\Symantec_Client_Security\Symantec AntiVirus\DefWatch.exe
svchost.exe	492	232	17	203	C:\WINNT\System32\svchost.exe
sqlservr.exe	536	232	30	246	
Rtvscan.exe	612	232	36	248	C:\Program Files\Symantec_Client_Security\Symantec AntiVirus\Rtvscan.exe
mstask.exe	648	232	7	121	C:\WINNT\system32\MSTask.exe
winmgmt.exe	676	232	16	309	C:\WINNT\System32\WBEM\WinMgmt.exe

svchost.exe	724	232	6	147	C:\WINNT\system32\svchost.exe
inetinfo.exe	752	232	21	343	C:\WINNT\System32\inetsrv\inetinfo.exe
mssearch.exe	792	232	8	160	C:\Program Files\Common Files\Microsoft Shared\MSSearch\Bin\mssearch.exe
explorer.exe	1132	1120	14	341	C:\WINNT\Explorer.EXE
VPTray.exe	1212	1132	4	130	C:\PROGRA~1\SYMANT~1\SYMANT~1\vptray.exe
CTFMON.EXE	1220	1132	1	71	C:\WINNT\System32\ctfmon.exe
sqlmangr.exe	552	1132	5	163	C:\Program Files\Microsoft SQL Server\80\Tools\Binn\sqlmangr.exe
wuauclt.exe	1028	724	6	114	C:\WINNT\System32\wuauclt.exe
ntvdm.exe	1400	1132	2	71	C:\WINNT\system32\ntvdm.exe
cmd.exe	132	1132	1	23	C:\WINNT\system32\cmd.exe
SecReport.exe	1156	132	5	131	E:\GetInfo.PUB\SecReport.exe

Page File settings

Pagefile name	Initial Size, MB	Maximum Size, MB
D:\pagefile.sys	512	512
C:\pagefile.sys	2	2

Hardware

Computer system

Brand: Intel
Model: S2440BX
Serial No.:
Number of processors: 1
BIOS Version: 4S4EB2X0.86A.0024.P17
BIOS Date: PhoenixBIOS 4.0 Release 6.0
RAM size, MBytes: 384

Processors

CPU ID	Manufacturer	Name	Max Speed, MHz	L2 Cache, KB	ExtClock, MHz
CPU0	GenuineIntel	Intel Pentium III processor	448	512	100

Logical Disks

Drive Letter	Description	FileSystem	Total Size, MB	Free Space, MB	Volume name	Serial No.
A:	3 1/2 Inch Floppy Drive	FAT	1	1	QSRWD_USR	8405903E
C:	Local Fixed Disk	NTFS	10001	6117		70269A7A
D:	Local Fixed Disk	FAT32	2959	2446	SWAP	887F82E4
E:	Local Fixed Disk	NTFS	16473	4170	SoftWare	BC9840C1
G:	CD-ROM Disc	CDFS	510	0	Forensic20030306	8061DB0B
I:	Removable Disk					
R:	CD-ROM Disc	CDFS	636	0	WinSecuring_2.1	908C656E

Physical Disks

Device ID	Model	Interface	Size, Bytes	Partitions	Bytes/Sector	Sec/Track	Cyl	Heads	Sectors	Tracks	Tr/Cyl
\\.\PHYSICALDRIVE0	WDC WD136AA	IDE	13596387840	2	512	63	1653	255	26555445	421515	255
\\.\PHYSICALDRIVE1	Maxtor 91728D8	IDE	17273088000	1	512	63	2100	255	33736500	535500	255

Mixed checkpoints

Recovery Console installed: True
Norton Antivirus signature date: 2003-03-10

Information collected with *SecReport.exe*, version *3.03.07*

Appendix 2.2. – List of files used during execution of SecReport tool

Note: Results of FileMon program, that monitored system during execution of tool SecReport, were:

1. exported into Microsoft Excel;
2. sorted by name of used file;
3. column with sorted list of used files saved as tab-delimited text file;
4. executed command:

```
uniq -i FileMonSecReportSorted4.txt > FileMonSortedUniq.txt
```

This command outputs only unique lines; option "`-i`" makes it case-insensitive. Below is complete list of unique filenames, used by SecReport tool that was obtained as result of processing described above:

C:\
C:\??\C:\WINNT\system32\cmd.exe
C:\autoexec.bat
C:\boot.ini
C:\cmdcons
C:\Documents and Settings\
C:\Documents and Settings\Administrator\
C:\Documents and Settings\Administrator\Local Settings\Temp
C:\Documents and Settings\All Users\Application Data
C:\Documents and Settings\All Users\Application Data\Microsoft
C:\Documents and Settings\All Users\Application Data\Microsoft\Crypto
C:\Documents and Settings\All Users\Application Data\Microsoft\Crypto\RSA
C:\Documents and Settings\All Users\Application Data\Microsoft\Crypto\RSA\MachineKeys
C:\Documents and Settings\All Users\Application Data\Microsoft\Crypto\RSA\MachineKeys\3501338eaa4ef8df5fbadf23ada5f36a_eb9aed62-6de3-4e57-8915-8b8fadb5dce2
C:\Tools\wshEN.DLL
C:\Tools\wshENU.DLL
C:\WINNT
C:\WINNT\
C:\WINNT\hh.exe
C:\WINNT\REPAIR\SETUP.LOG
C:\WINNT\system\wldap32.dll
C:\WINNT\system\wshEN.DLL
C:\WINNT\system\wshENU.DLL
C:\WINNT\system32
C:\WINNT\System32\
C:\WINNT\System32\~CLBCATQ.DLL
C:\WINNT\System32\ACTIVEDS.dll

C:\WINNT\system32\activeds.tlb
C:\WINNT\System32\ADMWPROX.DLL
C:\WINNT\system32\adsiis.dll
C:\WINNT\system32\adsldp.dll
C:\WINNT\System32\ADSLDPC.DLL
C:\WINNT\system32\adsmsext.dll
C:\WINNT\System32\adsnt.dll
C:\WINNT\System32\advapi32.dll
C:\WINNT\system32\aspperf.dll
C:\WINNT\system32\basesrv.dll
C:\WINNT\system32\browser.dll
C:\WINNT\System32\CLBCATQ.DLL
C:\WINNT\System32\cmd.exe
C:\WINNT\System32\cmd.exe.Local
C:\WINNT\system32\config\
C:\winnt\system32\config\appevent.evt
C:\winnt\system32\config\secevent.evt
C:\WINNT\system32\config\software.LOG
C:\winnt\system32\config\sysevent.evt
C:\WINNT\system32\crypt32.dll
C:\WINNT\system32\cryptdlg.dll
C:\WINNT\System32\CSCDLL.DLL
C:\WINNT\system32\cscript.exe
C:\WINNT\System32\cscui.dll
C:\WINNT\System32\DHCPCSVC.DLL
C:\WINNT\System32\DNSAPI.dll
C:\WINNT\system32\dnsrslvr.dll
C:\WINNT\system32\drivers\ksecdd.sys
C:\WINNT\system32\drivers\raspptp.sys
C:\WINNT\system32\drivers\rdpwd.sys
C:\WINNT\system32\drivers\srv.sys
C:\WINNT\system32\eventlog.dll
C:\WINNT\system32\FTPCTRS2.DLL
C:\WINNT\system32\gdi32.dll
C:\WINNT\System32\ICMP.DLL
C:\WINNT\system32\idq.dll
C:\WINNT\system32\iisrtl.dll
C:\WINNT\System32\INETSLOC.dll
C:\WINNT\system32\inetsrv\asp.dll
C:\WINNT\system32\inetsrv\fscfg.dll
C:\WINNT\system32\inetsrv\ftpmib.dll
C:\WINNT\system32\inetsrv\ftpsvc2.dll
C:\WINNT\system32\inetsrv\httpext.dll
C:\WINNT\system32\inetsrv\httpmib.dll
C:\WINNT\system32\inetsrv\httpodbc.dll
C:\WINNT\system32\inetsrv\IISLOG.DLL

C:\WINNT\system32\inetsrv\infocomm.dll
C:\WINNT\system32\inetsrv\ISATQ.DLL
C:\WINNT\system32\inetsrv\ism.dll
C:\WINNT\system32\inetsrv\mdsync.dll
C:\WINNT\system32\inetsrv\ssinc.dll
C:\WINNT\system32\inetsrv\urlscan\urlscan.dll
C:\WINNT\system32\inetsrv\w3svc.dll
C:\WINNT\system32\inetsrv\wam.dll
C:\WINNT\system32\INFOADMN.DLL
C:\WINNT\system32\infoctrs.dll
C:\WINNT\System32\iphlpapi.dll
C:\WINNT\system32\itss.dll
C:\WINNT\system32\jview.exe
C:\WINNT\system32\kdcsvc.dll
C:\WINNT\system32\kerberos.dll
C:\WINNT\system32\localspl.dll
C:\WINNT\system32\locator.exe
C:\WINNT\system32\lsass.exe
C:\WINNT\System32\MPRAPI.dll
C:\WINNT\system32\msdtcui.DLL
C:\WINNT\system32\msgina.dll
C:\WINNT\system32\mshtml.dll
C:\WINNT\System32\MSI.DLL
C:\WINNT\system32\msjava.dll
C:\WINNT\system32\msv1_0.dll
C:\WINNT\System32\MSVBVM60.DLL
C:\WINNT\system32\msw3prt.dll
C:\WINNT\System32\MSWSOCK.dll
C:\WINNT\System32\msxml3.dll
C:\WINNT\System32\msxml3r.dll
C:\WINNT\System32\NETAPI32.dll
C:\WINNT\system32\netlogon.dll
C:\WINNT\system32\netman.dll
C:\WINNT\System32\NETRAP.dll
C:\WINNT\System32\NETUI0.DLL
C:\WINNT\System32\NETUI1.DLL
C:\WINNT\system32\NTDLL.DLL
C:\WINNT\system32\ntdsa.dll
C:\WINNT\System32\ntlanman.dll
C:\WINNT\System32\OemInfo.Ini
C:\WINNT\System32\OemLogo.Bmp
C:\WINNT\System32\ole32.dll
C:\WINNT\system32\perfc009.dat
C:\WINNT\system32\Perfctrs.dll
C:\WINNT\system32\perfdisk.dll
C:\WINNT\system32\perfh009.dat

C:\WINNT\system32\Perflib_Perfdata_370.dat
C:\WINNT\system32\perfnet.dll
C:\WINNT\system32\perfos.dll
C:\WINNT\system32\perfproc.dll
C:\WINNT\system32\perfts.dll
C:\WINNT\system32\printui.dll
C:\WINNT\System32\PSAPI.DLL
C:\WINNT\system32\PSBASE.DLL
C:\WINNT\system32\query.dll
C:\WINNT\System32\rasadhlp.dll
C:\WINNT\System32\RASAPI32.DLL
C:\WINNT\System32\RASMAN.DLL
C:\WINNT\System32\RASSAPI.dll
C:\WINNT\System32\rnr20.dll
C:\WINNT\System32\rsabase.dll
C:\WINNT\System32\RTUTILS.DLL
C:\WINNT\System32\SAMLIB.dll
C:\WINNT\system32\samsrv.dll
C:\WINNT\system32\scecli.dll
C:\WINNT\system32\scesrv.dll
C:\WINNT\System32\ScrRun.dll
C:\WINNT\System32\Secur32.dll
C:\WINNT\System32\SETUPAPI.DLL
C:\WINNT\system32\shdocvw.dll
C:\WINNT\system32\shell32.dll
C:\WINNT\system32\sp3res.dll
C:\WINNT\system32\spoolss.dll
C:\WINNT\system32\srvsvc.dll
C:\WINNT\System32\stdole2.tlb
C:\WINNT\System32\TAPI32.DLL
C:\WINNT\system32\tapiperf.dll
C:\WINNT\system32\urlmon.dll
C:\WINNT\system32\user32.dll
C:\WINNT\System32\USERENV.DLL
C:\WINNT\System32\vbscript.dll
C:\WINNT\System32\vbscript.dll\3
C:\WINNT\system32\w32time.dll
C:\WINNT\system32\w32tm.exe
C:\WINNT\system32\W3CTRS.DLL
C:\WINNT\System32\WBEM\
C:\WINNT\System32\WBEM\aspperf.dll
C:\WINNT\System32\wbem\fastprox.dll
C:\WINNT\System32\WBEM\infoctrs.dll
C:\WINNT\System32\WBEM\Logs\FrameWork.log
C:\WINNT\System32\WBEM\msdtcui.DLL
C:\WINNT\System32\WBEM\Perfctrs.dll

C:\WINNT\System32\WBEM\perfdisk.dll
C:\WINNT\System32\WBEM\perfnet.dll
C:\WINNT\System32\WBEM\perfos.dll
C:\WINNT\System32\WBEM\perfproc.dll
C:\WINNT\System32\WBEM\perfts.dll
C:\WINNT\System32\WBEM\query.dll
C:\WINNT\System32\WBEM\tapiperf.dll
C:\WINNT\System32\WBEM\w3ctrs.dll
C:\WINNT\System32\wbem\wbemcomn.dll
C:\WINNT\System32\wbem\wbemdisp.dll
C:\WINNT\System32\wbem\wbemdisp.TLB
C:\WINNT\System32\wbem\wbemprox.dll
C:\WINNT\System32\wbem\wbemsvc.dll
C:\WINNT\System32\WBEM\winspool.drv
C:\WINNT\System32\Wbem\wshEN.DLL
C:\WINNT\System32\Wbem\wshENU.DLL
C:\WINNT\system32\win32spl.dll
C:\WINNT\system32\winlogon.exe
C:\WINNT\System32\winrnr.dll
C:\WINNT\System32\WINSPOOL.DRV
C:\WINNT\system32\wlnotify.dll
C:\WINNT\System32\WMI.dll
C:\WINNT\System32\WS2_32.dll
C:\WINNT\System32\WS2HELP.DLL
C:\WINNT\System32\wshEN.DLL
C:\WINNT\System32\wshENU.DLL
C:\WINNT\System32\wshom.ocx
C:\WINNT\System32\WSOCK32.dll
C:\WINNT\system32\xactsrv.dll
C:\WINNT\system32\xenroll.dll
C:\WINNT\TEMP
C:\WINNT\wshEN.DLL
C:\WINNT\wshENU.DLL

Appendix 2.3. – Sample report produced by Delta tool

Notes:

- If some value printed without / (slash character), that means that values are identical for System1 and System2 (In this report, for example, values "OS Version" are identical for both systems).
- If some values printed with "/" slash between them, that means that value before slash is for System1, value after slash – for System2. In this case, values are different. In example below, Media Player Version is different for systems.
- If some value has syntax:

`Value1 / -` that means that value exists only for System1 and does not exist for System2. In example below (Table "Services", entry "lanmanserver") service lanmanserver (and it's properties) exist only for System1.

- If some value has syntax:

`/ Value2 -` that means that value exists only for System2 and does not exist for System1. In example below (Table "Services", entry "DefWatch") service DefWatch (and it's properties) exist only for System2.

End of notes.

Differences between systems:

	System1	System2
Hostname	AK1	AK3
Date of report:	2003-03-08	2003-03-08
Time of report:	13:16:16	11:23:32

OS Version: Microsoft Windows XP Professional 5.1.2600
OS Service Pack: 1.0
IE Version: 6.0.2800.1106
Java VM Version: 5.0.3809.0
Media Player Version: 8.0.0.4487 And 6.4.9.1125 / 9.0.0.2980
WSH Version: 5.6.0.6626

Applications

Application	AK1	AK3
Cisco Systems VPN Client 3.6.3 (Rel) 3.6	Yes	No
CMD Prompt Here PowerToy	No	Yes
LiveUpdate 1.7 (Symantec Corporation)	No	Yes
Microsoft Visual Studio 6.0 Enterprise Edition	No	Yes

Microsoft Web Embedding Fonts Tool (III)	Yes	No
Microsoft Web Publishing Wizard 1.53	No	Yes
MP4DVD Video Decodec 1.0 Play release	No	Yes
MSDN Library - October 2001	No	Yes
Paint Shop Pro 7 7.0.2.0000	No	Yes
Panicware Pop-Up Stopper Companion	Yes	No
Run shell extension	No	Yes
Search and Replace	No	Yes
SpywareRemover 5.00.0000	No	Yes
Symantec AntiVirus Client 8.0.0.374	No	Yes
TextPad 4.6 4.6.0	Yes	No
TextPad 4.6 4.6.2	No	Yes
Tweakui Powertoy for Windows XP 1.00.0001	No	Yes
TweakVB	No	Yes
Window Washer	No	Yes
Windows Script V5.6 Documentation	No	Yes
WinZip 8.1 (4331)	Yes	No
WinZip 8.1 SR-1 (5266)	No	Yes

HotFixes

Hotfix	AK1	AK3
MS02-070 329170	No	Yes
MS03-001 810833	No	Yes
MS03-005 810577	No	Yes
MS03-004 810847	No	Yes

Services

Service	Start Type	Status	Account
AppMgmt	Manual/Disabled	Stopped/Stopped	LocalSystem/LocalSystem
BITS	Manual/Manual	Stopped/Running	LocalSystem/LocalSystem
Browser	Automatic/Disabled	Running/Stopped	LocalSystem/LocalSystem
CVPND/	Automatic/	Running/	LocalSystem/
FastUserSwitchingCompatibility	Disabled/Manual	Stopped/Stopped	LocalSystem/LocalSystem
ImapiService	Manual/Disabled	Stopped/Stopped	LocalSystem/LocalSystem

lanmanserver/	Automatic/	Running/	LocalSystem/
MSDTC	Manual/Disabled	Stopped/Stopped	NT AUTHORITY\NetworkService/NT AUTHORITY\NetworkService
PolicyAgent	Manual/Automatic	Stopped/Running	LocalSystem/LocalSystem
Spooler	Automatic/Manual	Running/Stopped	LocalSystem/LocalSystem
SSDPSRV	Manual/Disabled	Stopped/Stopped	NT AUTHORITY\LocalService/NT AUTHORITY\LocalService
TapiSrv	Manual/Automatic	Running/Running	LocalSystem/LocalSystem
TermService	Disabled/Manual	Stopped/Running	LocalSystem/LocalSystem
WebClient	Automatic/Manual	Running/Stopped	NT AUTHORITY\LocalService/NT AUTHORITY\LocalService
WmdmPmSp/	Automatic/	Running/	LocalSystem/
/DefWatch	/Disabled	/Stopped	/LocalSystem
/Norton AntiVirus Server	/Disabled	/Stopped	/LocalSystem
/WmdmPmSN	/Disabled	/Stopped	/LocalSystem

Ports

Port	Protocol
1028/	TCP/
3005/	TCP/
3247/	TCP/
3256/	TCP/
1029/	UDP/
3011/	UDP/
3019/	UDP/
3027/	UDP/
4500/	UDP/
62515/	UDP/
62517/	UDP/
62519/	UDP/
62521/	UDP/
62523/	UDP/
62524/	UDP/

/1026	/UDP
/3009	/UDP
/3054	/UDP
/3117	/UDP
/3453	/UDP
/3006	/UDP
/3012	/UDP
/3018	/UDP
/3089	/UDP

Processes

Process name	AK1	AK3
spoolsv.exe	Yes	No
cvpnd.exe	Yes	No
ScreenSeize.exe	Yes	No
DefWatch.exe	No	Yes
Rtvscan.exe	No	Yes
VPTray.exe	No	Yes
PopUpWatch.exe	No	Yes
SpyWatch.exe	No	Yes
washer.exe	No	Yes
AcroTray.exe	No	Yes
WINCMD32.EXE	No	Yes
cmd.exe	No	Yes

References

[1] Hal91 – Floppy Linux - : http://jspiro.tripod.com/linux/hal91.htm

[2] GCC Releases – Timeline - http://gcc.gnu.org/releases.html

[3] FreshRPMs.net – RedHat - http://freshrpms.net/redhat.html

[4] DistroWatch.com – Red Hat Linux - http://www.distrowatch.com/table.php?distribution=redhat

[5] daemon9 - "LOKI2 (The Implementation)" - online Magazine "Phrack", Vol.7, Issue 51, Article 06 (published: September 01, 1997) - http://www.phrack.com/phrack/51/P51-06

[6] daemon9 – "[Project Loki]" - online Magazine "Phrack", Vol.7, Issue 49, Article 06 (published: September 01, 1997) - http://www.phrack.org/phrack/49/P49-06

[7] RFC 792 - Internet Control Message Protocol. Published: September 1981 - http://www.faqs.org/rfcs/rfc792.html

[8] U.S. Department of Defense - DOD 5200.28-STD - Trusted Computer System Evaluation Criteria – December 1985 - http://www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.html#HDR8

[9] Java Security – Security threats – Covert channels - http://ei.cs.vt.edu/~wwwbtb/book/chap14/twothree.html

[10] Kimball, Aaron - Covert Channels in HTTP 1.1 - http://www.concentric.net/~kimballa/covert.html

[11] Craig H. Rowland - Covert channels in the TCP/IP protocol suite - http://www.firstmonday.dk/issues/issue2_5/rowland/

[12] Mixter - A brief programming tutorial in C for raw sockets - http://mixter.void.ru/rawip.txt

[13] RPMfind.net - libc-5.3.12-31 RPM for i386 - http://rpmfind.net//linux/RPM/redhat/6.2/i386/libc-5.3.12-31.i386.html

[14] NetAdminTools.com – Linux ABCs - atd http://www.netadmintools.com/modules.php?name=News&new_topic=35

[15] United States Code (U.S.C.) § 2701 – Unlawful access to storage communications - http://www.cybercrime.gov/ECPA2701_2712.htm

[16] United States Code (U.S.C.) § 1030 – Fraud and related activity in connection with computers - http://www.cybercrime.gov/1030NEW.htm

[17] Rob Lee – SANS course material: Investigative incident response and basic forensic Windows principles – Hands On. – SANS Institute, 2002.

[18] Kevin Mandia, Chris Prosise – Incident Response. Investigating computer crime. – Osborne/McGraw Hill, 2001 – Chapter 9: Initial respone to Windows NT/2000, pp. 225-252.

[19] Resource Kit for Windows 2000 – Auditpol - http://www.dynawell.com/support/ResKit/win2k.asp

[20] FPort v.2.0 – Foundstone – Free Tools – Intrusion Detection Tools – http://www.foundstone.com/knowledge/intrusion_detection.html

[21] Microsoft Baseline Security Analyzer http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/Security/tools/tools/MBSAHome.ASP

[22] Mssecure.xml – file with database on Micosoft hotfixes. Download: http://download.microsoft.com/download/xml/security/1.0/NT5/EN-US/mssecure.cab . This file automatically downloaded from Microsoft site when tool SecReport.exe executed with parameter "-l", like: `SecReport.exe -l`

[23] Resource Kit for Windows 2000 – Forfiles - http://www.dynawell.com/support/ResKit/win2k.asp

[24] MSXSL.EXE Command Line Transformation Utility http://msdn.microsoft.com/downloads/default.asp?URL=/downloads/sample.asp?url=/msdn-files/027/001/485/msdncompositedoc.xml

[25] Markus Kuhn - A Summary of the International Standard Date and Time Notation - http://www.cl.cam.ac.uk/~mgk25/iso-time.html

[26] Microsoft Network Security Hotfix Checker (Hfnetchk.exe) Tool Is Available – Microsoft TechNet article 303215 - http://support.microsoft.com/default.aspx?scid=kb;en-us;303215&sd=tech

[27] Urlscan Security Tool – MicroSoft TechNet article - http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/tools/locktool.asp

[28] Symantec AntiVirus Corporate Edition http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=155&EID=0

[29] MSDN: Microsoft Visual J# .NET - The New Microsoft Java Virtual Machine - http://msdn.microsoft.com/vjsharp/productinfo/visualj/downloads/wfcinfo.asp

[30] SysInternals – Utilities – PsTools – PsList - http://www.sysinternals.com/ntw2k/freeware/pslist.shtml

[31] Microsoft MSDN - Platform SDK: Windows Management Instrumentation - Win32_Process - http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wmisdk/wmi/win32_process.asp

[32] GNU.org – software –Textutils – md5sum - http://www.gnu.org/software/textutils/textutils.html

[33] DependencyWalker.com – Dependency Walker for x86 - http://www.dependencywalker.com/

[34] SysInternals – Utilities – Filemon for Windows - http://www.sysinternals.com/ntw2k/source/filemon.shtml

[35] NTSecurity.com (web site of Arne Vidstrom) – Toolbox – Macmatch - http://www.ntsecurity.nu/toolbox/macmatch/

[36] Symantec Ghost™ Corporate Edition 7.5 - http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=3

[37] PowerQuest – Products – DriveImage - http://www.powerquest.com/driveimage/

[38] SysInternals – Utilities – Monitoring Tools – Process Explorer http://www.sysinternals.com/ntw2k/freeware/procexp.shtml

[39]. 18 U.S.C. 2703 - Requirements for Governmental Access. http://www.cybercrime.gov/usc2703.htm

[40] Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations – Section III-G - Working with Network Providers: Preservation of Evidence, Preventing Disclosure to Subjects, and Cable Act Issues - http://www.cybercrime.gov/s&smanual2002.htm#_IIIG1_

[41] Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations APPENDIX C: Sample Language for Preservation Request Letters under 18 U.S.C. § 2703(f)
http://www.cybercrime.gov/s&sappendix2002.htm#_C_

[42] 18 U.S.C. 2701 - Unlawful Access to Stored Communications
http://www.cybercrime.gov/usc2701.htm

[43] Computer Crime and Intellectual Property Section (CCIPS) – How to Report Internet-Related Crime
http://www.cybercrime.gov/reporting.htm

[44] United States Code Annotated - Title 18. Crimes And Criminal Procedure - Part I – Crimes - Chapter 47 - Fraud And False Statements
http://www.cybercrime.gov/1030NEW.htm

[45] McKinney's Consolidated Laws Of New York Annotated Penal Law. Chapter 40 Of The Consolidated Laws Part Three--Specific Offenses - Title J--Offenses Involving Theft Article 156--Offenses Involving Computers; Definition Of Terms
http://unitedstates.datastar.net/newyork.htm

[46] Creator of Melissa Computer Virus Sentenced to 20 Months in Federal Prison
http://www.usdoj.gov/criminal/cybercrime/melissaSent.htm

[47] Russian Computer Hacker Sentenced To Three Years In Prison
http://www.usdoj.gov/criminal/cybercrime/gorshkovSent.htm

[48] London, England Hacker Indicted Under Computer Fraud and Abuse Act For Accessing military computer.
http://www.usdoj.gov/criminal/cybercrime/mckinnonIndict.htm

[49] Uniting And Strengthening America By Providing Appropriate Tools Required To Intercept And Obstruct Terrorism (USA Patriot Act) Act Of 2001
http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107_cong_public_laws&docid=f:publ056.107.pdf

[50] The Cyber Security Enhancement Act - Provisions of Section 225 of the Homeland Security Act of 2002
http://www.cybercrime.gov/homeland_CSEA.htm

[51] Amendments & Redline Showing Changes Resulting From Section 896 Of The 2002 Homeland Security Act And Section Sec. 225 Of The 2002 Cyber Security Enhancement Act
http://www.cybercrime.gov/homeland_225.htm