



## **Systems and Network Programming**

**2020 Regular Intake**

**Title:**

**PHP Remote Code Execution Attack CVE-2019-11043**

**IT19127910**

**R.P.A.Ranasinghe**

## **Content**

## **Page**

Introduction	03
Who found the vulnerability?	04
How to exploit this vulnerability	05
Conclusion	12
References	13

## Introduction



Some versions of php7 which run on NGINX with php-fpm enabled may be vulnerable to remote code execution cve-2019-11043. The vulnerability has to do with a lack of testing on the NGINX and PHP-FPM configurations. The vulnerability can be exploited to achieve remote execution code under some conditions.

PHP-FPM is the use of PHP FastCGI, which provides a progressive and unique product ready for content written in PHP programming language. PHP-FPM is not a core component of NGINX installations and usually integrates with web hosting providers into their PHP environment.

Andrew Danau, a security researcher at Wallarm, discovered the vulnerability in the Capture the Flag match, and was later trained by two of his fellow researchers, Omar Ganiv and Emil Lerner, for a fully functional remote code execution exploit.

The main problem is an overflow memory corruption problem in the PHP-FPM module called "env\_path\_info" which allows attackers to execute arbitrary code remotely on vulnerable web servers.

## **Who found the vulnerability?**

Wallarm security researcher, Andrew Danau stumbled on an odd conduct of PHP script when taking part in Capture The Flag (CTF), which took place September 14–16, 2019.

The server response was peculiar when Andrew Danau sent 0% (newline) byte in the URL. It gets more data back than it should be there. And, the sum of additional data was related to the number of bytes inside the URL after 0%.

Typically, this kind of response is linked to memory corruption attacks and we expected an assault on the kind of disclosure of the information. Disclosure of information is bad enough because it may lead to the leaking of confidential or financial data. Even worse, from time to time, though such behaviour can rarely indicate a vulnerability in the execution of remote code.

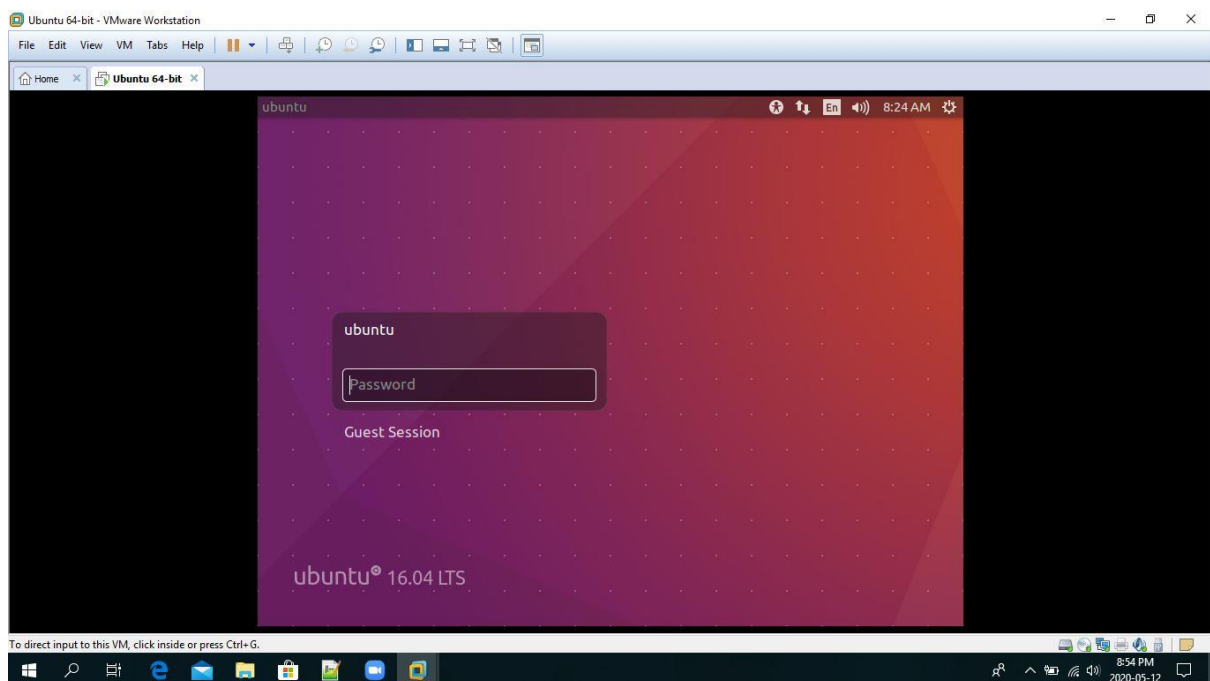
## How to exploit this vulnerability?

We need to install

- Python 3.6.1
- Php version 7
- Nginx

In our Linux operating system.

This bug was found in 2019. The attacker must have a Linux version on or before year 2018. So, I had to install a virtual box in with Ubuntu OS 16.4.



First, download python from <https://www.python.org/downloads/source/>

- Python 3.6.1 - March 21, 2017
  - Download Gzipped source tarball
  - Download XZ compressed source tarball

Then, go in to the downloaded python folder and start installing the Python.

```
# cd Downloads
```

```
# cd Python-3.6.1
```

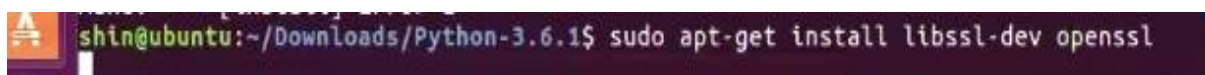
```
# ./configure
```



```
Terminal
shin@ubuntu: ~/Downloads/Python-3.6.1
shin@ubuntu:~/Downloads$ cd ..
shin@ubuntu:~$ cd Downloads
shin@ubuntu:~/Downloads$ ls
nginx_1.14.0.orig.tar.gz  Python-3.6.1  Python-3.6.1.tgz
shin@ubuntu:~/Downloads$ cd Python-3.6.1
shin@ubuntu:~/Downloads/Python-3.6.1$ ls
aclocal.m4  Doc  LICENSE  Objects  pyconfig.h.in
config.guess  Grammar  Mac  Parser  Python
config.sub  Include  Makefile.pre.in  PC  README.rst
configure  install-sh  Misc  PCbuild  setup.py
configure.ac  Lib  Modules  Programs  Tools
shin@ubuntu:~/Downloads/Python-3.6.1$ ./configure
checking build system type...
```

After the configuration is complete, install Python.

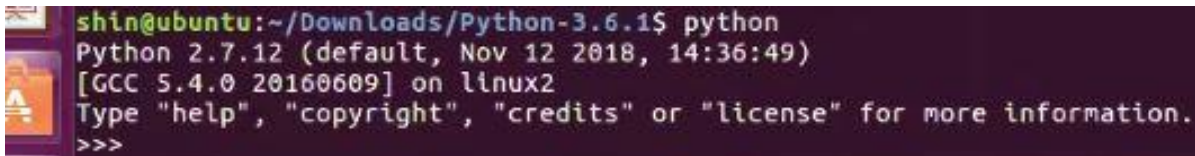
```
# sudo apt-get install libssh-dev openssl
```



```
shin@ubuntu:~/Downloads/Python-3.6.1$ sudo apt-get install libssh-dev openssl
```

After installing run python to check weather it is running properly.

# python

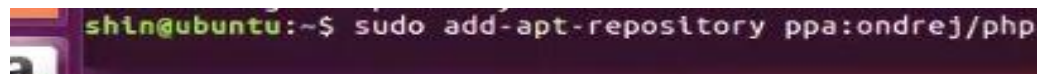
A terminal window with a dark background. The prompt is 'shin@ubuntu:~/Downloads/Python-3.6.1\$'. The user has entered 'python'. The output shows 'Python 2.7.12 (default, Nov 12 2018, 14:36:49) [GCC 5.4.0 20160609] on linux2'. It then prompts 'Type "help", "copyright", "credits" or "license" for more information.' followed by '>>>' on the next line.

```
shin@ubuntu:~/Downloads/Python-3.6.1$ python
Python 2.7.12 (default, Nov 12 2018, 14:36:49)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Then, install PHP.

To install PHP, add php repositories

# sudo add-apt-repository ppa:ondrej/php

A terminal window showing the command 'sudo add-apt-repository ppa:ondrej/php' being entered at the prompt 'shin@ubuntu:~\$'.

```
shin@ubuntu:~$ sudo add-apt-repository ppa:ondrej/php
```

After that update apt

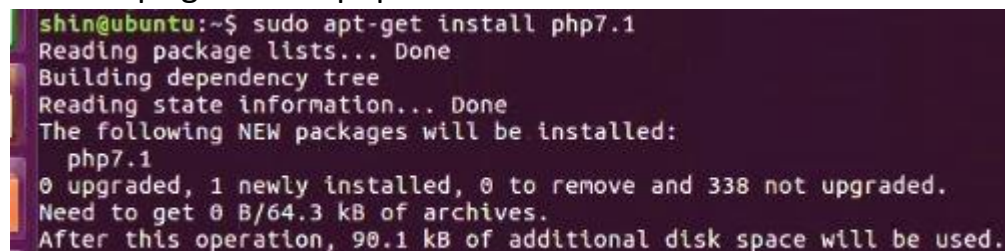
# sudo apt-get update

A terminal window showing the output of 'sudo apt-get update'. It lists two hits: 'http://ppa.launchpad.net/ondrej/php/ubuntu xenial InRelease' and 'http://security.ubuntu.com/ubuntu xenial-security InRelease'. It also shows progress for 'gpgv' at 0% and a message 'Waiting for headers'.

```
shin@ubuntu:~$ sudo apt-get update
Hit:1 http://ppa.launchpad.net/ondrej/php/ubuntu xenial InRelease
Hit:2 http://security.ubuntu.com/ubuntu xenial-security InRelease
0% [2 InRelease gpgv 109 kB] [Waiting for headers]
```

Install php

# sudo apt-get install php7.1

A terminal window showing the output of 'sudo apt-get install php7.1'. It lists the steps: 'Reading package lists... Done', 'Building dependency tree', and 'Reading state information... Done'. It then lists the new packages to be installed: 'php7.1'. It also shows the disk space requirements: '0 upgraded, 1 newly installed, 0 to remove and 338 not upgraded. Need to get 0 B/64.3 kB of archives. After this operation, 90.1 kB of additional disk space will be used.'

```
shin@ubuntu:~$ sudo apt-get install php7.1
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  php7.1
0 upgraded, 1 newly installed, 0 to remove and 338 not upgraded.
Need to get 0 B/64.3 kB of archives.
After this operation, 90.1 kB of additional disk space will be used.
```



```
shin@ubuntu:~$ sudo apt-get install php7.1-interbase php7.1-intl php7.1-json php7.1-ldap php7.1-mbstring php7.1-mcrypt php7.1-mysql php7.1-odbc php7.1-openssl php7.1-redis php7.1-sqlite3 php7.1-ssh2 php7.1-tidy php7.1-xmlrpc php7.1-xsl php7.1-zip
php7.1-interbase - Interbase module for PHP
php7.1-intl - Internationalisation module for PHP
php7.1-json - JSON module for PHP
php7.1-ldap - LDAP module for PHP
php7.1-mbstring - MBSTRING module for PHP
php7.1-mcrypt - libmcrypt module for PHP
php7.1-mysql - MySQL module for PHP
php7.1-odbc - ODBC module for PHP
php7.1-openssl - Zend OpenSSL module for PHP
php7.1-redis - Redis module for PHP
php7.1-ssh2 - SSH2 module for PHP
php7.1-sqlite3 - SQLite3 module for PHP
php7.1-sybase - Sybase module for PHP
php7.1-tidy - tidy module for PHP
php7.1-xml - DOM, SimpleXML, WDDX, XML, and XSL module for PHP
php7.1-xmlrpc - XMLRPC-EPI module for PHP
php7.1-xsl - XSL module for PHP (dummy)
php7.1-zip - Zip module for PHP
shin@ubuntu:~$ sudo apt-get
```

After that ,

# sudo nano /etc/php/7.1/cli/php.ini

```
shin@ubuntu:~$ sudo nano /etc/php/7.1/cli/php.ini
sudo: nano /etc/php/7.1/cli/php.ini: command not found
shin@ubuntu:~$ sudo nano /etc/php/7.1/cli/php.ini

GNU nano 2.5.3 File: /etc/php/7.1/cli/php.ini Modified

;cgi.nph = 1

; if cgi.force_redirect is turned on, and you are not running under Apache or NS
; (iPlanet) web servers, you MAY need to set an environment variable name that $
; will look for to know it is OK to continue execution. Setting this variable $
; cause security issues, KNOW WHAT YOU ARE DOING FIRST.
; http://php.net/cgi.redirect-status-env
;cgi.redirect_status_env =

; cgi.fix_pathinfo provides *real* PATH_INFO/PATH_TRANSLATED support for CGI. $
; previous behaviour was to set PATH_TRANSLATED to SCRIPT_FILENAME, and to not $
; what PATH_INFO is. For more information on PATH_INFO, see the cgi specs. Se$
; this to 1 will cause PHP CGI to fix its paths to conform to the spec. A sett$
; of zero causes PHP to behave as before. Default is 1. You should fix your s$
; to use SCRIPT_FILENAME rather than PATH_TRANSLATED.
; http://php.net/cgi.fix-pathinfo
;cgi.fix_pathinfo=0

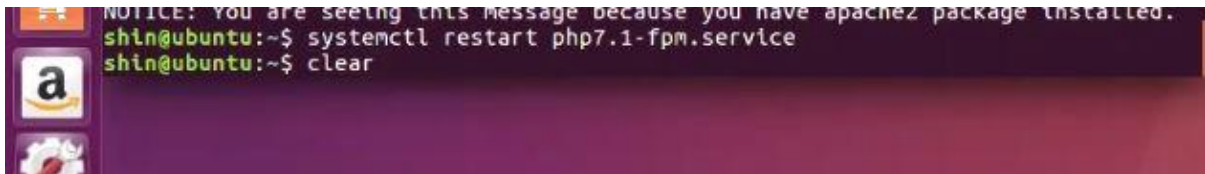
; if cgi.disable_path is enabled, the PHP CGI binary can safely be placed outs$

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```



Then restart the php

```
# systemctl restart php7.1-fpm.service
```

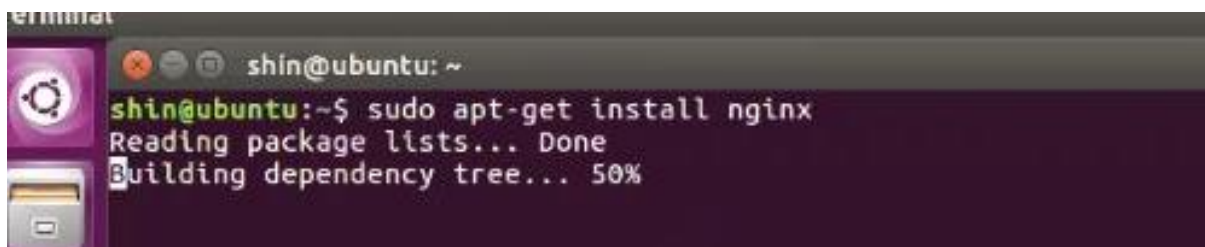
A terminal window with a dark background and light text. The prompt is 'shin@ubuntu:~\$'. The user has entered 'systemctl restart php7.1-fpm.service' and 'clear'. A notice at the top says 'NOTICE: you are seeing this message because you have apache2 package installed.'

```
shin@ubuntu:~$ systemctl restart php7.1-fpm.service
shin@ubuntu:~$ clear
```

In third,

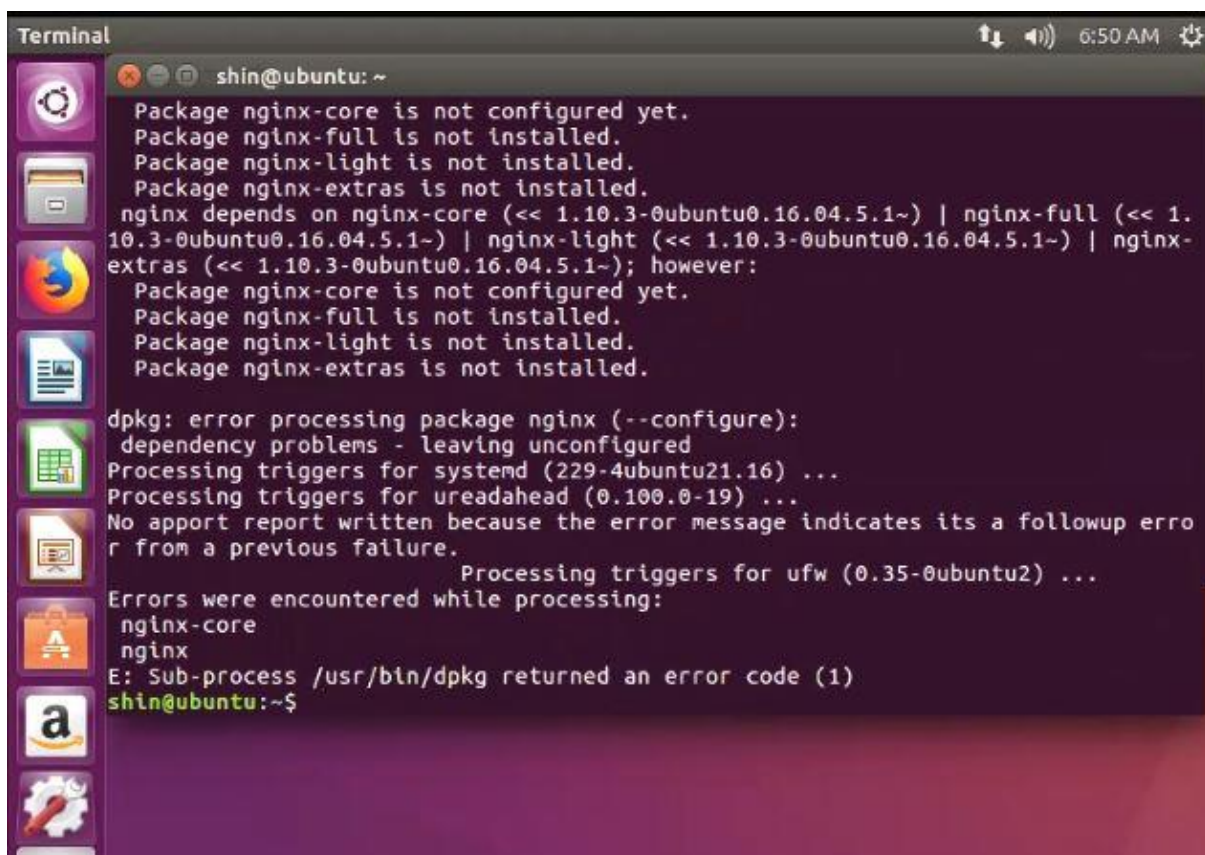
Install nginx

```
# sudo apt-get install nginx
```

A terminal window showing the command 'sudo apt-get install nginx' being executed. The output shows 'Reading package lists... Done' and 'Building dependency tree... 50%'.

```
shin@ubuntu:~$ sudo apt-get install nginx
Reading package lists... Done
Building dependency tree... 50%
```

But when I was trying to install nginx, the result was this..

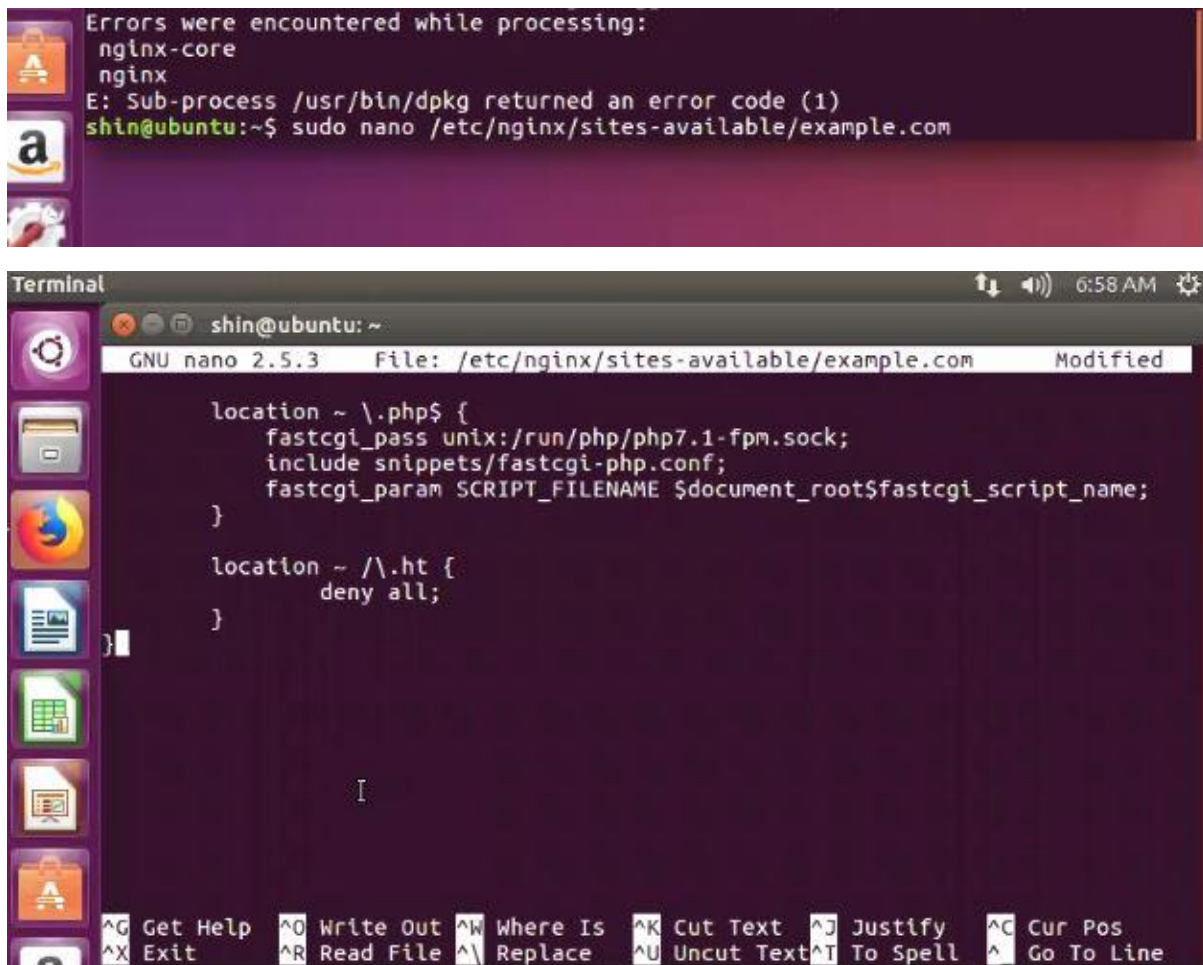
A terminal window showing the error message when installing nginx. The output indicates that nginx depends on nginx-core, nginx-full, nginx-light, or nginx-extras, which are not installed or configured. The installation fails with an error code (1).

```
Package nginx-core is not configured yet.
Package nginx-full is not installed.
Package nginx-light is not installed.
Package nginx-extras is not installed.
nginx depends on nginx-core (< 1.10.3-0ubuntu0.16.04.5.1~) | nginx-full (< 1.10.3-0ubuntu0.16.04.5.1~) | nginx-light (< 1.10.3-0ubuntu0.16.04.5.1~) | nginx-extras (< 1.10.3-0ubuntu0.16.04.5.1~); however:
Package nginx-core is not configured yet.
Package nginx-full is not installed.
Package nginx-light is not installed.
Package nginx-extras is not installed.

dpkg: error processing package nginx (--configure):
dependency problems - leaving unconfigured
Processing triggers for systemd (229-4ubuntu21.16) ...
Processing triggers for ureadahead (0.100.0-19) ...
No apport report written because the error message indicates its a followup error from a previous failure.
Processing triggers for ufw (0.35-0ubuntu2) ...
Errors were encountered while processing:
 nginx-core
 nginx
E: Sub-process /usr/bin/dpkg returned an error code (1)
shin@ubuntu:~$
```

Then

#sudo nano etc/nginx/sites-available/example.com



The image shows a terminal window with a dark purple background. At the top, it displays an error message: "Errors were encountered while processing: nginx-core nginx". Below this, it says "E: Sub-process /usr/bin/dpkg returned an error code (1)". The prompt is "shin@ubuntu:~\$ sudo nano /etc/nginx/sites-available/example.com". Below the terminal window, there is a screenshot of the nano text editor. The title bar shows "GNU nano 2.5.3" and the file path "/etc/nginx/sites-available/example.com". The editor content shows two location blocks: one for ".php" files using fastcgi\_pass and include snippets, and another for ".ht" files denying all access. The bottom status bar of the nano editor lists various keyboard shortcuts like ^G Get Help, ^O Write Out, etc.

```
Errors were encountered while processing:
nginx-core
nginx
E: Sub-process /usr/bin/dpkg returned an error code (1)
shin@ubuntu:~$ sudo nano /etc/nginx/sites-available/example.com

GNU nano 2.5.3 File: /etc/nginx/sites-available/example.com Modified

location ~ /\.php$ {
    fastcgi_pass unix:/run/php/php7.1-fpm.sock;
    include snippets/fastcgi-php.conf;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
}

location ~ /\.ht {
    deny all;
}

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

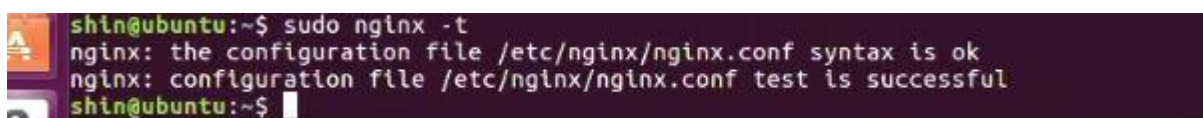
# sudo ln -s /etc/nginx/sites-available/examples.com/etc/nginx/sites-enabled/example.com



The image shows a terminal window with a dark purple background. The prompt is "shin@ubuntu:~\$". The user enters "sudo nano /etc/nginx/sites-available/example.com" twice. Then, the user enters "sudo ln -s /etc/nginx/sites-available/example.com/etc/nginx/sites-enabled/example.com". The prompt returns to "shin@ubuntu:~\$".

```
shin@ubuntu:~$ sudo nano /etc/nginx/sites-available/example.com
shin@ubuntu:~$ sudo nano /etc/nginx/sites-available/example.com
shin@ubuntu:~$ sudo ln -s /etc/nginx/sites-available/example.com/etc/nginx/sites-enabled/example.com
shin@ubuntu:~$
```

# sudo nginx -t

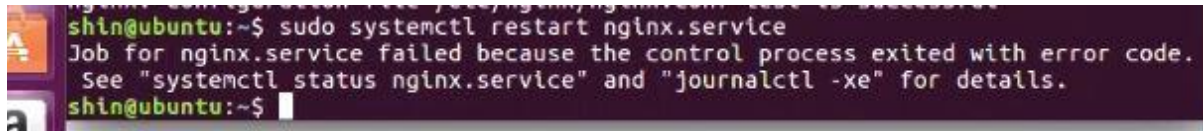


The image shows a terminal window with a dark purple background. The prompt is "shin@ubuntu:~\$". The user enters "sudo nginx -t". The output shows "nginx: the configuration file /etc/nginx/nginx.conf syntax is ok" and "nginx: configuration file /etc/nginx/nginx.conf test is successful". The prompt returns to "shin@ubuntu:~\$".

```
shin@ubuntu:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
shin@ubuntu:~$
```

From here I was unable to continue the installing and run nginx successfully.

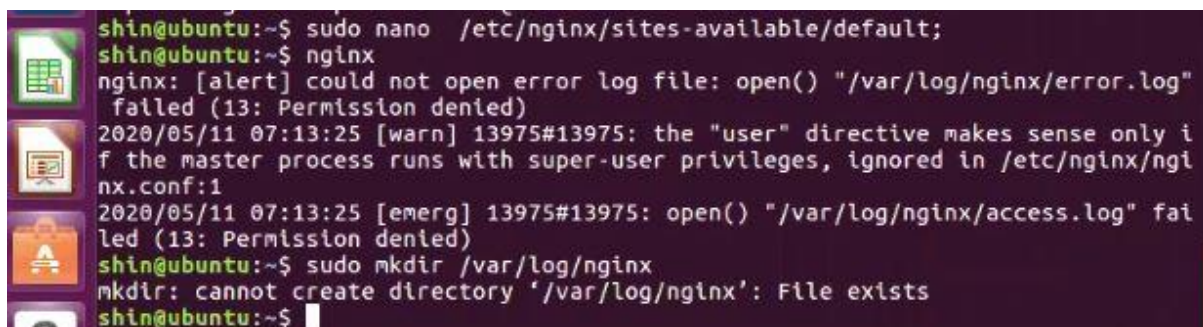
# sudo systemctl restart nginx.service

A terminal window showing a failed command. The user runs 'sudo systemctl restart nginx.service'. The output indicates the job failed because the control process exited with an error code. It suggests checking 'systemctl status nginx.service' and 'journalctl -xe' for details.

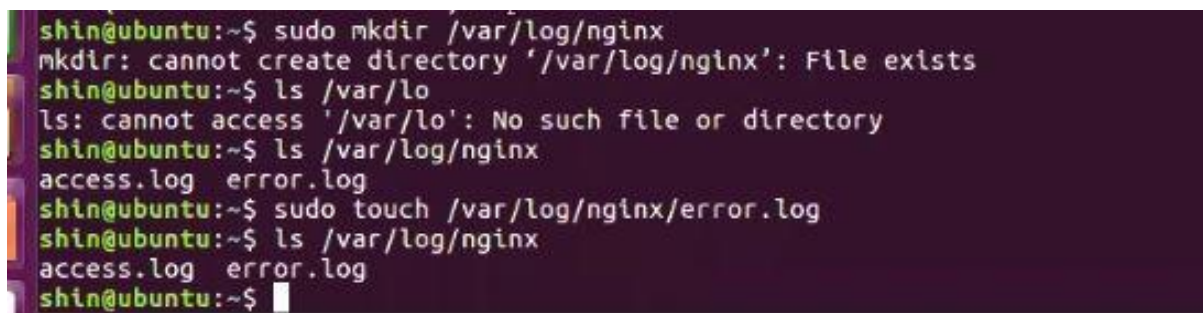
```
shin@ubuntu:~$ sudo systemctl restart nginx.service
Job for nginx.service failed because the control process exited with error code.
See "systemctl status nginx.service" and "journalctl -xe" for details.
shin@ubuntu:~$
```

# sudo nano /etc/nginx/sites-available/default;

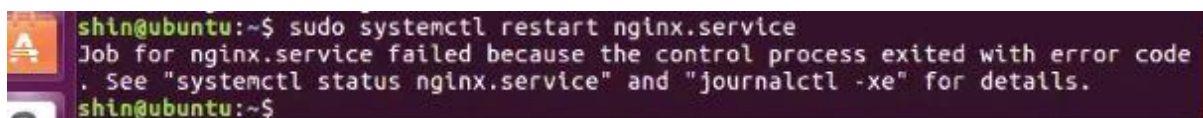
#nginx

A terminal window showing the user running 'sudo nano /etc/nginx/sites-available/default;' and then 'nginx'. The output shows several error messages: 'nginx: [alert] could not open error log file: open() "/var/log/nginx/error.log" failed (13: Permission denied)', a warning about the 'user' directive, and '2020/05/11 07:13:25 [emerg] 13975#13975: open() "/var/log/nginx/access.log" failed (13: Permission denied)'. The user then runs 'sudo mkdir /var/log/nginx', which fails with 'mkdir: cannot create directory "/var/log/nginx": File exists'.

```
shin@ubuntu:~$ sudo nano /etc/nginx/sites-available/default;
shin@ubuntu:~$ nginx
nginx: [alert] could not open error log file: open() "/var/log/nginx/error.log"
failed (13: Permission denied)
2020/05/11 07:13:25 [warn] 13975#13975: the "user" directive makes sense only i
f the master process runs with super-user privileges, ignored in /etc/nginx/ngi
nx.conf:1
2020/05/11 07:13:25 [emerg] 13975#13975: open() "/var/log/nginx/access.log" fai
led (13: Permission denied)
shin@ubuntu:~$ sudo mkdir /var/log/nginx
mkdir: cannot create directory '/var/log/nginx': File exists
shin@ubuntu:~$
```

A terminal window showing the user running 'sudo mkdir /var/log/nginx', which fails with 'mkdir: cannot create directory "/var/log/nginx": File exists'. The user then runs 'ls /var/lo', which fails with 'ls: cannot access "/var/lo": No such file or directory'. Next, the user runs 'ls /var/log/nginx', which shows 'access.log error.log'. The user then runs 'sudo touch /var/log/nginx/error.log', which succeeds. Finally, the user runs 'ls /var/log/nginx', which shows 'access.log error.log'.

```
shin@ubuntu:~$ sudo mkdir /var/log/nginx
mkdir: cannot create directory '/var/log/nginx': File exists
shin@ubuntu:~$ ls /var/lo
ls: cannot access '/var/lo': No such file or directory
shin@ubuntu:~$ ls /var/log/nginx
access.log  error.log
shin@ubuntu:~$ sudo touch /var/log/nginx/error.log
shin@ubuntu:~$ ls /var/log/nginx
access.log  error.log
shin@ubuntu:~$
```

A terminal window showing the user running 'sudo systemctl restart nginx.service'. The output indicates the job failed because the control process exited with an error code. It suggests checking 'systemctl status nginx.service' and 'journalctl -xe' for details.

```
shin@ubuntu:~$ sudo systemctl restart nginx.service
Job for nginx.service failed because the control process exited with error code
. See "systemctl status nginx.service" and "journalctl -xe" for details.
shin@ubuntu:~$
```

Due to this error situation I was unable to continue my exploiting.

## Conclusion

Sometimes there is a huge positive coming out of thinking off-course, or rather being a pioneer of new courses to solve shared challenges. In this case, Andrew Danaou was able to contribute to the community; making space for the web applications more secure, with a timely accidental discovery of a zero-day vulnerability that was unknown to the CTF designer or anyone in the community. PHP is responsible for many modern websites, including popular web platforms such as WordPress and Drupal. While the team that maintains PHP is diligent, it is important to quickly patch newly identified vulnerabilities such as this one and an earlier RCE, described in CVE-2019-13224.

By installing updated security patches, updating the software to the latest updates, we can prevent this vulnerability.



## References

Digitalocean.com. 2020. *How To Install Linux, Nginx, Mysql, PHP (LEMP Stack) In Ubuntu 16.04 | Digitalocean*. [online] Available at:

<<https://www.digitalocean.com/community/tutorials/how-to-install-linux-nginx-mysql-php-lemp-stack-in-ubuntu-16-04>> [Accessed 12 May 2020].

GitHub. 2020. *Vulhub/Vulhub*. [online] Available at:

<<https://github.com/vulhub/vulhub/blob/master/php/CVE-2019-11043/README.md>> [Accessed 12 May 2020].

Kumar, M., 2020. *New PHP Flaw Could Let Attackers Hack Sites Running On Nginx Servers*. [online] The Hacker News. Available at:

<<https://thehackernews.com/2019/10/nginx-php-fpm-hacking.html>> [Accessed 12 May 2020].

Sarva, S., 2020. *PHP Remote Code Execution Vulnerability (CVE-2019-11043) | Qualys Blog*. [online] Qualys Blog. Available at:

<<https://blog.qualys.com/webappsec/2019/10/30/php-remote-code-execution-vulnerability-cve-2019-11043>> [Accessed 12 May 2020].

Medium. 2020. *PHP-FPM Remote Code Execution Vulnerability (CVE-2019-11043)*

*Analysis*. [online] Available at: <<https://medium.com/@knownsec404team/php-fpm-remote-code-execution-vulnerability-cve-2019-11043-analysis-35fd605dd2dc>> [Accessed 12 May 2020].