Systems and Network Programming

2020 Regular Intake

Title:

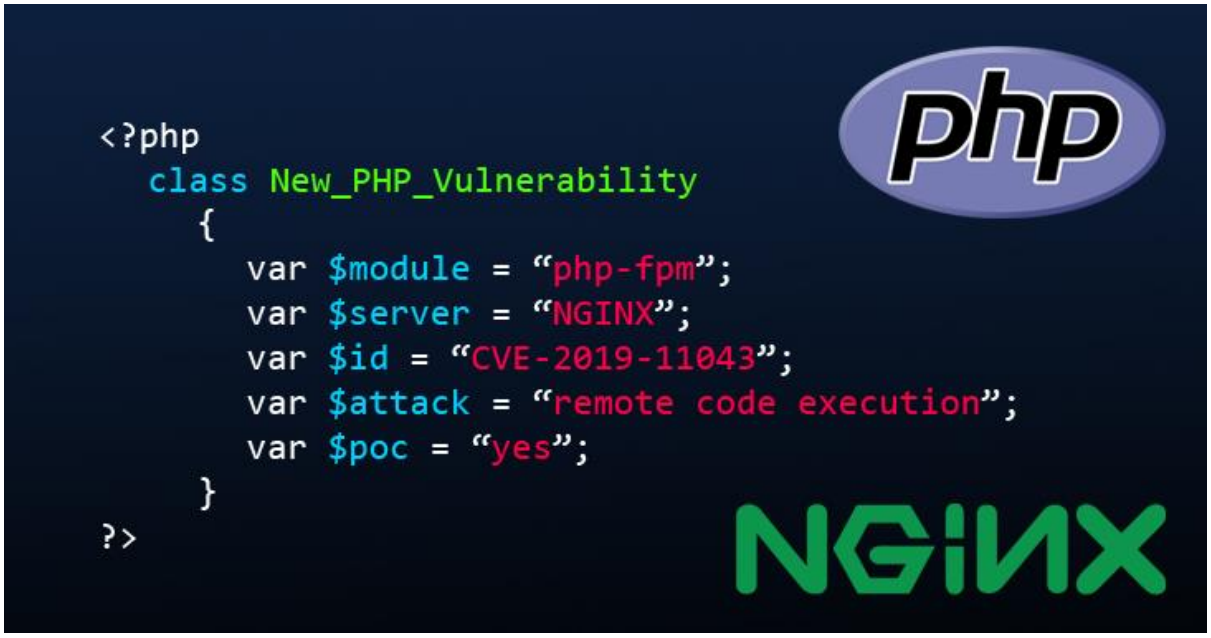
PHP Remote Code Execution Attack CVE-2019-11043

IT19127910

R.P.A.Ranasinghe

Content | Page

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.

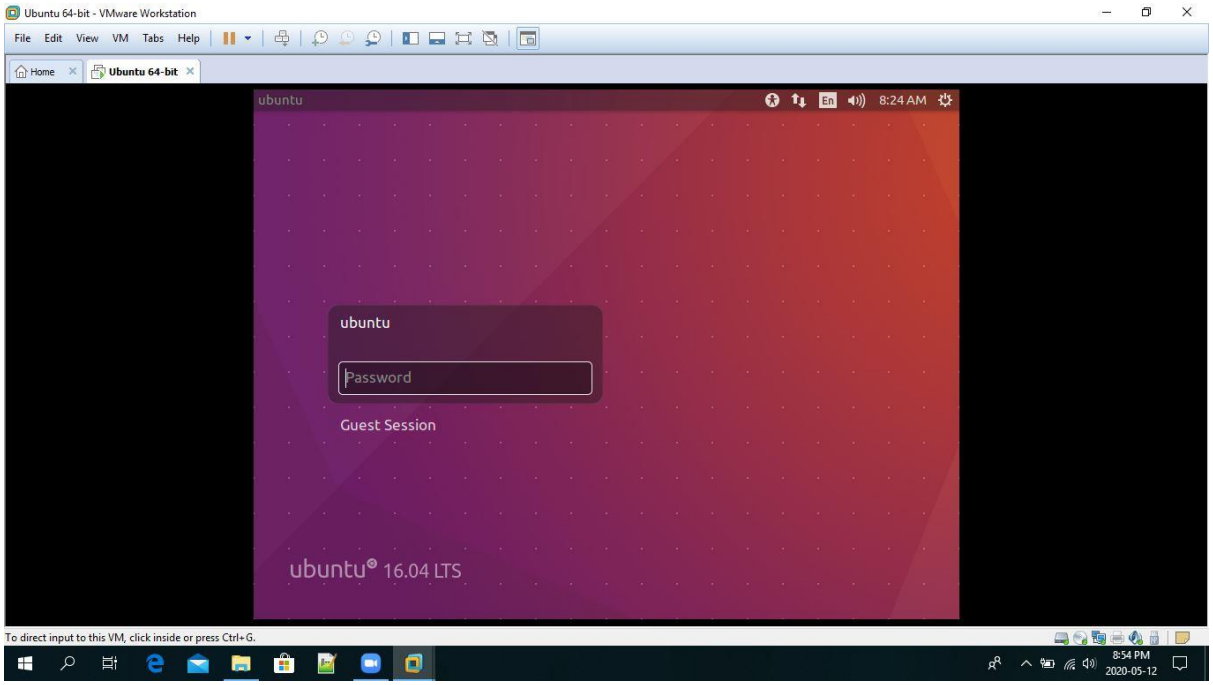
How to exploit this vulnerability?

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.



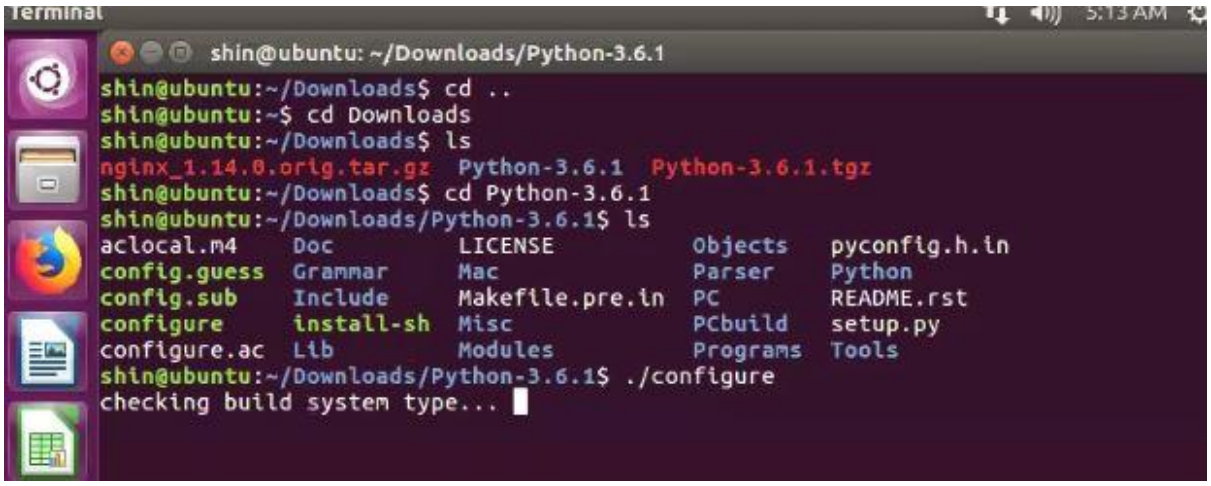
First, download python from https://www.python.org/downloads/source/

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

cd Downloads

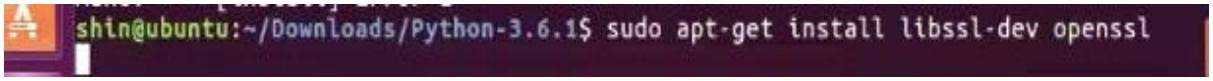
cd Python-3.6.1

./configure



After the configuration is complete, install Python.

sudo apt-get install libssh-dev openssl



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

python

```
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:ondre/php

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

After that update apt

sudo apt-get update

```
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

```
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.
```

```
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-opcache - Zend OpCache module for PHP
php7.1-pgsql - PostgreSQL module for PHP
php7.1-phpdbg - server-side, HTML-embedded scripting language (PHPDBG binary)
php7.1-pspell - pspell module for PHP
php7.1-readline - readline module for PHP
php7.1-recode - recode module for PHP
php7.1-snmp - SNMP module for PHP
php7.1-soap - SOAP 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-ge
```

After that ,

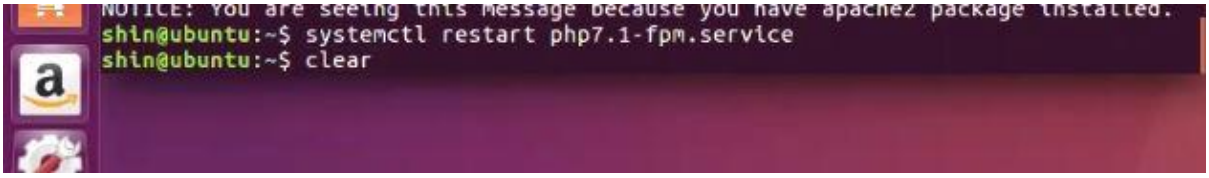
sudo nano /etc/php/7.1/cli/ini

```
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
```

```
shin@ubuntu: ~
  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 N$
; (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.discard_path is enabled, the PHP CGI binary can safely be placed outsi$

^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

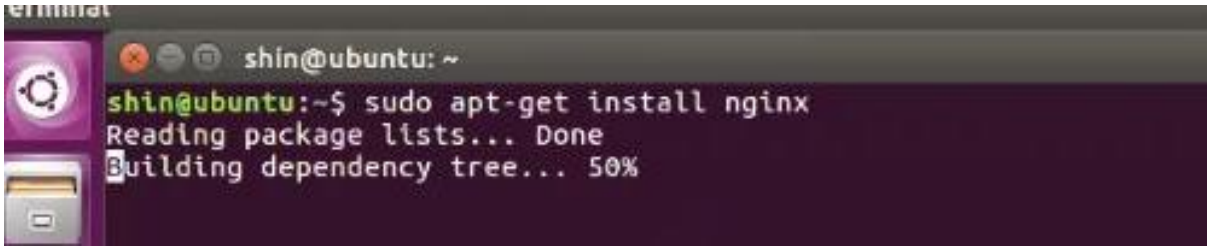
systemctl restart php7.1-fpm.service



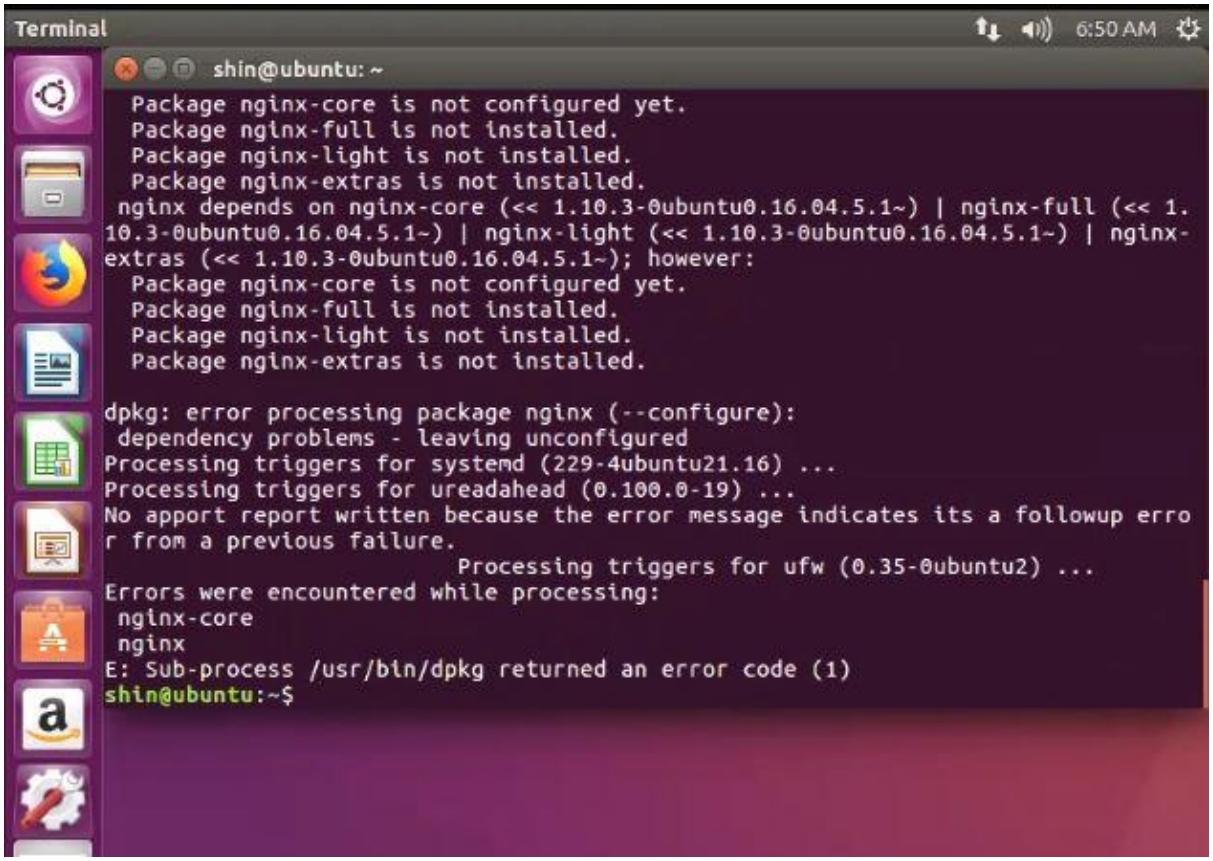
In third,

Install nginx

sudo apt-get install nginx

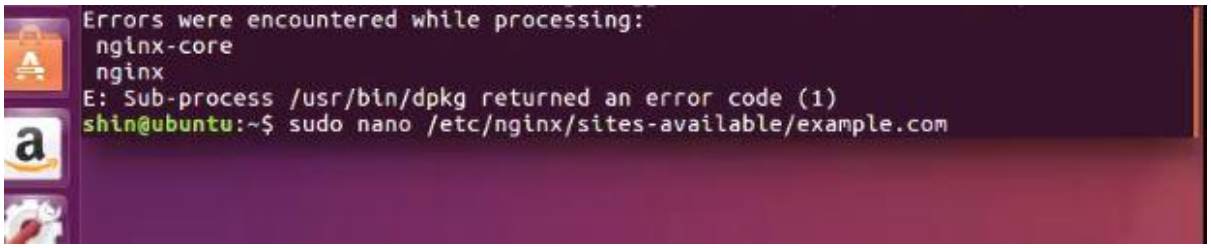


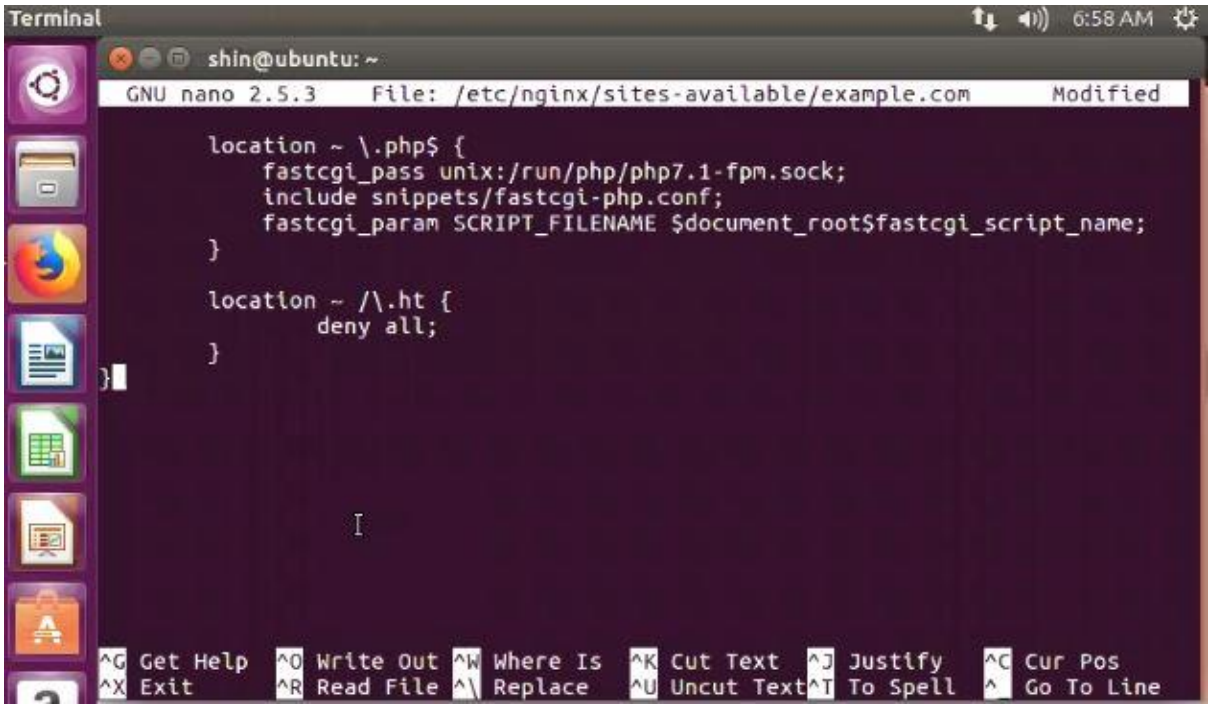
But when I was trying to install nginx, the result was this..



Then

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





sudo ln -s /etc/nginx/sites-avilable/examples.com/etc/nginx/sties-enabled/example.com

sudo nginx -t

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

sudo systemctl restart nginx.service

```
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-avilable/default;

#nginx

```
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:~$
```

```
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:~$
```

```
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 Danau 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].