

Blog



If you want to find vulnerabilities and make a money bug bounty hunting you may want to get a copy of my book. **BUY NOW!**

Exposed Log and Configuration Files

Slack Group

Before we get started I have started a slack group dedicated to hacking. We welcome everyone from beginner to advanced to join. I will be on everyday answer questions, doing CTFs, and talking about cool hacks. If you enjoy hacking and are looking for like minded people join below:

NEW Hacking Group Slack Channel

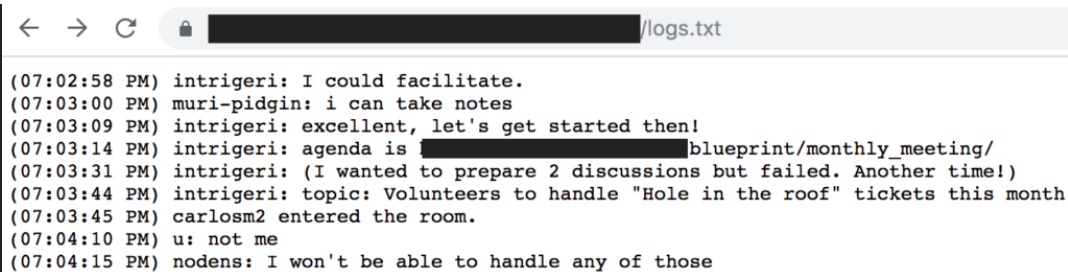
Introduction

Log and configuration files are used by applications to store error messages, warning messages, startup variables, and other things. These files can contain sensitive information such as passwords, encryption keys, tokens, and anything else you would want as an attacker. For applications that are not exposed to the internet this may not be a problem as no one would have access to the sensitive data but web applications are exposed to the internet.

Exposed Files

The only way to find exposed log and configuration files on an application is to know the file name. The best way to find these files is to create a list of common files and perform directory brute forcing. Some common log file names include:

- logs.txt
- log.txt
- debug.log
- *.log
- *.config
- etc

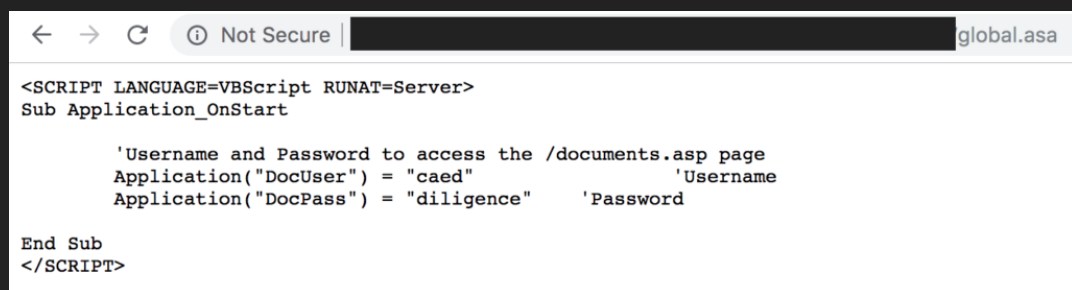
A screenshot of a web browser window displaying a chat log. The address bar shows a URL ending in /logs.txt. The log contains several messages with timestamps and usernames, including 'intrigeri', 'muri-pidgin', and 'carlosm2'.

```
(07:02:58 PM) intrigeri: I could facilitate.
(07:03:00 PM) muri-pidgin: i can take notes
(07:03:09 PM) intrigeri: excellent, let's get started then!
(07:03:14 PM) intrigeri: agenda is [REDACTED]blueprint/monthly_meeting/
(07:03:31 PM) intrigeri: (I wanted to prepare 2 discussions but failed. Another time!)
(07:03:44 PM) intrigeri: topic: Volunteers to handle "Hole in the roof" tickets this month
(07:03:45 PM) carlosm2 entered the room.
(07:04:10 PM) u: not me
(07:04:15 PM) nodens: I won't be able to handle any of those
```

You may be able to come up with a fairly good list of possible file names but a lot of application and frameworks name their files something like *“juio_life-framework.log”* which is not easily guessable. The only way to find these files is to understand what each technology stack and framework names their log and configuration files. There are thousands of different frameworks so I obviously can't talk about each one of them but ill go over a few so you get the idea.

Global.asa

The **global.asa** configuration file is used by ASP.net applications to store information and objects used by the application. These essential act as global variables. This file is typically stored in the root directory but that doesn't mean you wont find this file elsewhere. If you stumble across this file make sure to look for sensitive information such as usernames, passwords, and database credentials.

A screenshot of a web browser window displaying the content of a file named global.asa. The browser's address bar shows the file name. The content is a VBScript code block that defines a sub-routine Application_OnStart, which sets a username 'caed' and a password 'diligence' for a document named 'documents.asp'.

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
Sub Application_OnStart

    'Username and Password to access the /documents.asp page
    Application("DocUser") = "caed"      'Username
    Application("DocPass") = "diligence"  'Password

End Sub
</SCRIPT>
```

As you can see the above example contains a username with its associated password. Note that this file was not found on the root directory which is where I normally find these.

Larvel Framework


Larvel is a fairly popular PHP framework. This framework stores its logs files at *“/storage/logs/laravel.log”* which is accessible by anyone.

```
← → ↻ ⓘ Not Secure | [redacted]storage/logs/laravel.log






[2017-06-16 10:08:52] local.ERROR: Symfony\Component\Console\Exception\CommandNotFoundException: Command "routes/" is not defined.

Did you mean one of these?
  route:cache
  route:clear
  route:list in /var/www/html/webeo/vendor/symfony/console/Application.php:556
Stack trace:
#0 /var/www/html/webeo/vendor/symfony/console/Application.php(197): Symfony\Component\Console\Application->find('routes/')
#1 /var/www/html/webeo/vendor/symfony/console/Application.php(124): Symfony\Component\Console\Application->doRun(Object(Symfony\Component\Console\Input\ArgvInput), Object(Symfony\Component\Console\Output\ConsoleOutput))
>doRun(Object(Symfony\Component\Console\Input\ArgvInput), Object(Symfony\Component\Console\Output\ConsoleOutput))
```

This file can contain all kinds of hidden gems. You can find usernames, passwords, environment variables and more.

 **Sergey Bobrov (bobrov)**

6231 **87th** **6.38** **94th** **16.00** **84th**
Reputation Rank Signal Percentile Impact Percentile

56 **#401098** **[mena.starbucks.com] Laravel App Log & Configuration Disclosure.** Share:     

State **Resolved (Closed)**

Disclosed **September 30, 2019 2:38pm -0400**


Reported To **Starbucks**

Asset **Other assets (Other)**

Weakness **Information Disclosure**


Bounty **\$500**

Severity **High (7 ~ 8.9)**

Participants 

Visibility **Disclosed (Limited)**

Summary by Starbucks

 bobrov discovered a misconfiguration in a Laravel instance at mena.starbucks.com, which exposed log files and environment variables containing database management credentials. The logs have been removed, and the instance of Laravel has been disabled.

Thank you @bobrov for finding this misconfiguration and helping to resolve this issue!

Looks like Starbucks paid \$500 because someone found database credentials in this log file. That's \$500 dollars for something that probably took 60 seconds to find. That could easily be you.

Zend Framework

This is another framework that stores sensitive information in its configuration files. The `"/application/configs/application.ini"` file is used to store database credentials.

```
← → ↻ ⓘ Not Secure | [redacted]/application.ini

[production]
phpSettings.display_startup_errors = 0
phpSettings.display_errors = 0
includePaths.library = APPLICATION_PATH "/../library"
bootstrap.path = APPLICATION_PATH "/Bootstrap.php"
bootstrap.class = "Bootstrap"
appnamespace = "Application"
resources.frontController.controllerDirectory = APPLICATION_PATH "/controllers"
resources.frontController.params.displayExceptions = 0
resources.db.adapter = "pdo_mysql"
resources.db.param.host = "localhost"
resources.db.isDefaultTableAdapter = true
resources.db.params.username = "web146-w00007-pr"
resources.db.params.password = "dBaDmln2"
```

This file is not exposed by default but developers often mess things up and exposed this file by accident.

Conclusion

Finding exposed log and configuration files is one of the easiest ways to find credentials and other sensitive information. You can create a list of possible file names such as *log.txt* but most frameworks use a unique name. The only way to find these file names is to understand the framework and how it names its log or configuration files. This comes with experience and exposure to different technology stacks.

Filed under: [configuration files](#), [exposed files](#), [log files](#), [sensitive information](#)

[← Broken Link Hijacking](#)

[Google Exposed Firebase Database](#)

Your email address will not be published. Required fields are marked *



Post Comment

TWITTER



SLACK CHANNEL



GITHUB

