

# HIGH-PERFORMANCE WEB APPLICATION FINGERPRINTING

BASED ON SCM REPOSITORIES



András Veres-Szentkirályi 2017-11-18

# \$ whoami



## **András Veres-Szentkirályi**

- dnet
- OSCP, GWAPT, SISE
- co-founder of Silent Signal and H.A.C.K.
- pentester, toolmaker
- Hacker camp enthusiast

# We all know that feeling



A screenshot of the phpMyAdmin web interface in a browser window. The address bar shows 'http://10.13.37.42/phpmyadmin/'. The page features the phpMyAdmin logo (a sailboat) and the text 'Welcome to phpMyAdmin'. Below this is a 'Language' section with a dropdown menu currently set to 'English'. Further down is a 'Log in' section with a help icon, 'Username:' and 'Password:' labels, and corresponding input fields. A 'Go' button is located at the bottom right of the login section.

← ⓘ http://10.13.37.42/phpmyadmin/ 🔍 Search

  
**Welcome to phpMyAdmin**

**Language**

English ▼

**Log in** ⓘ

**Username:**

**Password:**

**Go**

## Phpmyadmin » Phpmyadmin » 4.4.6 : Security Vulnerabilities

Cpe Name: *cpe:/a:phpmyadmin:phpmyadmin:4.4.6*

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

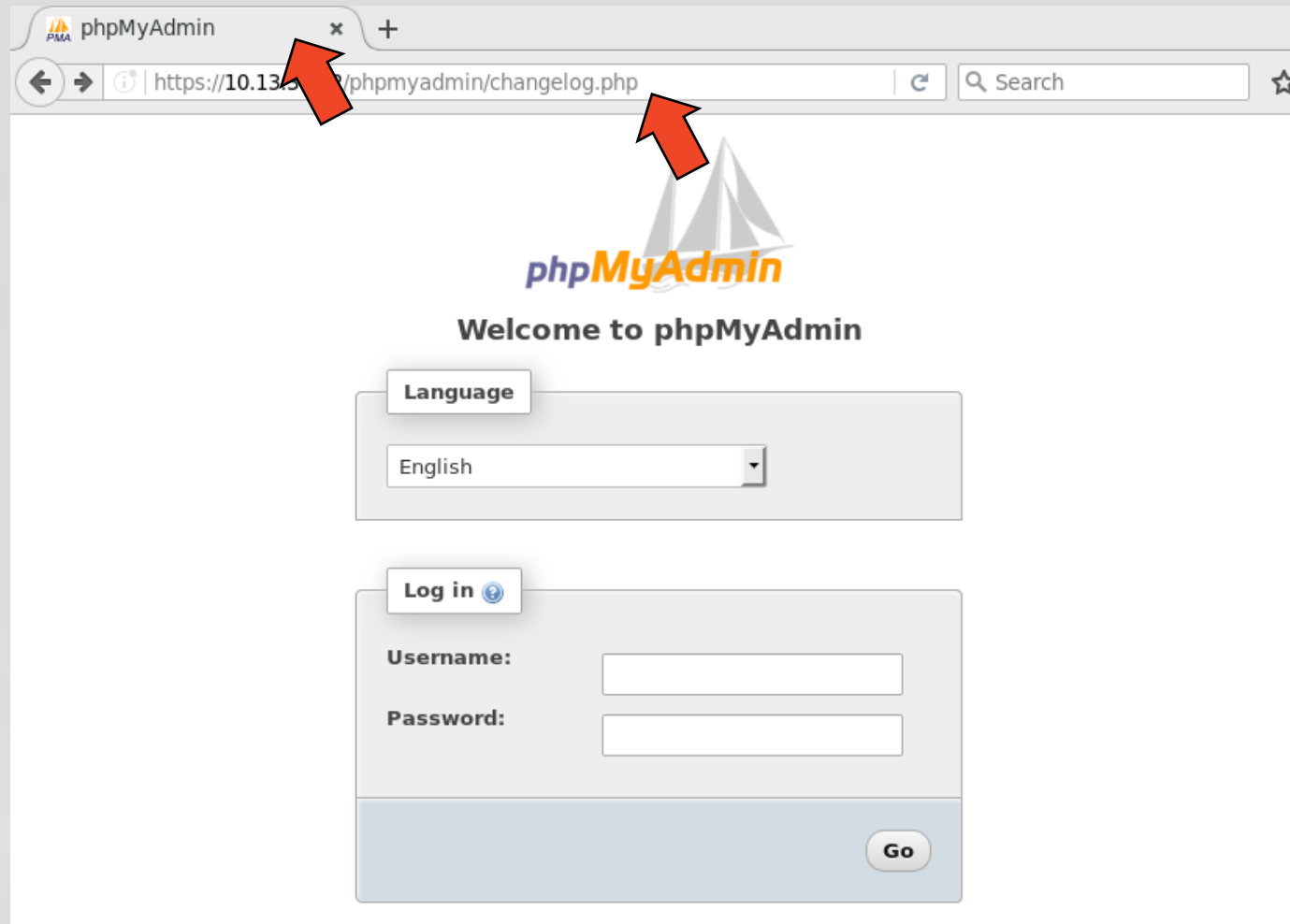
Sort Results By : [CVE Number Descending](#) [CVE Number Ascending](#) [CVSS Score Descending](#) [Number Of Exploits Descending](#)

[Copy Results](#) [Download Results](#)

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	<a href="#">CVE-2017-1000018</a>	<a href="#">20</a>			2017-07-17	2017-07-19	5.0	None	Remote	Low	Not required	None	None	Partial
phpMyAdmin 4.0, 4.4., and 4.6 are vulnerable to a DOS attack in the replication status by using a specially crafted table name														
2	<a href="#">CVE-2017-1000017</a>	<a href="#">918</a>			2017-07-17	2017-07-19	6.5	None	Remote	Low	Single system	Partial	Partial	Partial
phpMyAdmin 4.0, 4.4 and 4.6 are vulnerable to a weakness where a user with appropriate permissions is able to connect to an arbitrary MySQL server														
3	<a href="#">CVE-2017-1000015</a>	<a href="#">79</a>		XSS	2017-07-17	2017-07-19	4.3	None	Remote	Medium	Not required	None	Partial	None
phpMyAdmin 4.0, 4.4, and 4.6 are vulnerable to a CSS injection attack through crafted cookie parameters														
4	<a href="#">CVE-2017-1000014</a>	<a href="#">20</a>			2017-07-17	2017-07-19	5.0	None	Remote	Low	Not required	None	None	Partial
phpMyAdmin 4.0, 4.4, and 4.6 are vulnerable to a DOS weakness in the table editing functionality														
5	<a href="#">CVE-2017-1000013</a>	<a href="#">601</a>			2017-07-17	2017-07-19	5.8	None	Remote	Medium	Not required	Partial	Partial	None
phpMyAdmin 4.0, 4.4, and 4.6 are vulnerable to an open redirect weakness														
6	<a href="#">CVE-2016-9866</a>	<a href="#">352</a>		CSRF	2016-12-10	2017-06-30	6.8	None	Remote	Medium	Not required	Partial	Partial	Partial
An issue was discovered in phpMyAdmin. When the arg_separator is different from its default & value, the CSRF token was not properly stripped from the return URL of the preference import action. All 4.6.x versions (prior to 4.6.5), 4.4.x versions (prior to 4.4.15.9), and 4.0.x versions (prior to 4.0.10.18) are affected.														
7	<a href="#">CVE-2016-9865</a>	<a href="#">502</a>		Bypass	2016-12-10	2017-06-30	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
An issue was discovered in phpMyAdmin. Due to a bug in serialized string parsing, it was possible to bypass the protection offered by PMA_safeUnserialize() function. All 4.6.x versions (prior to 4.6.5), 4.4.x versions (prior to 4.4.15.9), and 4.0.x versions (prior to 4.0.10.18) are affected.														
8	<a href="#">CVE-2016-9864</a>	<a href="#">89</a>		Sql	2016-12-10	2017-06-30	6.0	None	Remote	Medium	Single system	Partial	Partial	Partial

An issue was discovered in phpMyAdmin. With a crafted username or a table name, it was possible to inject SQL statements in the tracking functionality that would run with the privileges of the control user. This gives read and write access to the tables of the configuration storage database, and if the control user has the necessary privileges, read access to some tables of the MySQL database. All 4.6.x versions (prior to 4.6.5), 4.4.x versions (prior to 4.4.15.9), and 4.0.x versions (prior to 4.0.10.18) are affected.

# WTF



# Use the source



```
➤ curl -s https://10.13.37.42/phpmyadmin/ | egrep -o '<(script|img|link)[^>]{,80}>?'  
<link rel="icon" href="favicon.ico" type="image/x-icon" />  
<link rel="shortcut icon" href="favicon.ico" type="image/x-icon" />  
<link rel="stylesheet" type="text/css" href="phpmyadmin.css.php?server=1&lang=en&  
<link rel="stylesheet" type="text/css" href="./themes/pmahomme/jquery/jquery-ui-1.9.2  
<script type='text/javascript' src='js/whitelist.php?lang=en&db=&collation_conn  
<script type="text/javascript" src="js/get_scripts.js.php?lang=en&collation_connect  
<script type='text/javascript' src='js/messages.php?lang=en&db=&collation_conne  
<script type='text/javascript' src='js/get_image.js.php?theme=pmahomme'>  
<script type="text/javascript">  
  
<img src="themes/dot.gif" title="Documentation" alt="Documentation" class="icon ic_b
```

# The basic idea



Release	SHA(file1.png)	SHA(file2.css)	SHA(file3.js)
v1.0	997b0bf3...	9d055618...	5df90ee1...
v1.1	997b0bf3...	b2e5ff43...	5df90ee1...
v1.2	997b0bf3...	b2e5ff43...	53718853...
v2.0	acf04ef6...	b2e5ff43...	5df90ee1...
v2.1	ef116a09...	b2e5ff43...	53718853...
v2.1.1	ef116a09...	b2e5ff43...	5df90ee1...
v3.0	ef116a09...	6f5681cf...	5df90ee1...
v4.0	ef116a09...	6f5681cf...	5df90ee1...
v4.0.1	ef116a09...	6f5681cf...	53718853...





**Isaac Wolkerstorfer**

@agnoster

Follow



Replying to @wilshiple

@wilshiple git gets easier once you get the basic idea that branches are homeomorphic endofunctors mapping submanifolds of a Hilbert space.

9:52 PM - 6 Mar 2011

584 Retweets 398 Likes



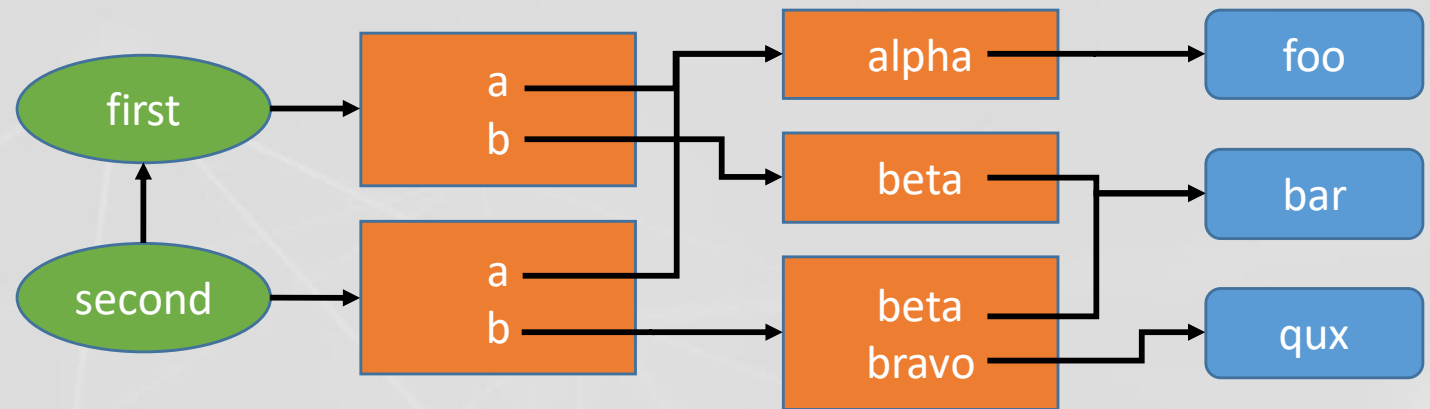


# Git internals






```
git init
mkdir a b
echo foo >a/alpha
echo bar >b/beta
git add .
git commit -m first

echo qux >b/bravo
git add .
git commit -m second
```



## GIT OBJECT TYPES

- Tag – points to a commit
- Commit – points to a tree 
- Tree – points to a list of trees and/or blobs 
- Blob – file contents 

# Git internals



```
foo ➤ git show --pretty=raw 0b2c018 | grep tree
tree 7fe7893039e138b720acb999ab4c6df1737bfcc3
foo ➤ git ls-tree 7fe7893039e138b720acb999ab4c6df1737bfcc3
040000 tree 283c5baf40ace28b21e3db7df2dc0517ecb2ff8a      a
040000 tree 19b50acca521b41857bde95faa775c922eb9e060      b
foo ➤ git ls-tree 19b50acca521b41857bde95faa775c922eb9e060
100644 blob 5716ca5987cbf97d6bb54920bea6adde242d87e6      beta
100644 blob 100b0dec8c53a40e4de7714b2c612dad5fad9985      bravo
foo ➤ echo -ne 'blob 4\0bar\n' | sha1sum
5716ca5987cbf97d6bb54920bea6adde242d87e6  -
```

**WHAT IF THE PATH  
DOESN'T EVEN MATTER**

**SINCE THE HASH OF THE  
BLOB ITSELF IS ENOUGH**

# Implementation



- Burp Suite → Java ☹
  - Pro not required!
- Git library: JGit (<https://www.eclipse.org/jgit/>, BSD licensed)
- Decoupled design
  - UI and logic separated
  - Standalone console version
  - Swing GUI when invoked as a Burp plugin



# Demo time!



# What could be better



- Which files are good candidates for further narrowing the set?
  - Burp can issue HTTP requests
  - Why not do it automatically?
- List relevant tags/versions
  - For those poor souls that don't know about `git-describe(1)`
- More user friendly UI
  - “Scratched my own itch”
  - WORKSFORME™
    - (OK, I'd be able to use better)
- Needs a cool name and a logo

# JVM vs. static calls



```
private static RecursiveResult isHashInTree(TreeWalk treeWalk, AnyObjectId tree,
                                             ObjectId hash, Set<AnyObjectId> knownToContain,
                                             Set<AnyObjectId> knownToBeFreeOf) throws IOException {
    ...
    if (treeWalk.isSubtree()) {
        treeWalk.enterSubtree();
        rr = isHashInTree(treeWalk, mid, hash, knownToContain, knownToBeFreeOf);
    }
}
```

**Result: more than 8 times better throughput**



# Method name of the week



## **disposeBody**

```
public final void disposeBody()
```

Discard the message buffer to reduce memory usage.

After discarding the memory usage of the RevCommit is reduced to only the `getTree()` and `getParents()` pointers and the time in `getCommitTime()`. Accessing other properties such as `getAuthorIdent()`, `getCommitterIdent()` or either message function requires reloading the buffer by invoking `RevWalk.parseBody(RevObject)`.

**Since:**

4.0

# Eager vs. lazy evaluation



SHA(file1.png)	SHA(file2.css)	SHA(file3.js)
997b0bf3...	9d055618...	5df90ee1...
997b0bf3...	b2e5ff43...	5df90ee1...
997b0bf3...	b2e5ff43...	53718853...
acf04ef6...	b2e5ff43...	5df90ee1...
ef116a09...	b2e5ff43...	53718853...
ef116a09...	b2e5ff43...	5df90ee1...
ef116a09...	6f5681cf...	5df90ee1...
ef116a09...	6f5681cf...	5df90ee1...
ef116a09...	6f5681cf...	53718853...

SHA(file1.png)	SHA(file2.css)	SHA(file3.js)
997b0bf3...	9d055618...	5df90ee1...
997b0bf3...	b2e5ff43...	5df90ee1...
997b0bf3...	b2e5ff43...	53718853...
acf04ef6...	b2e5ff43...	5df90ee1...
ef116a09...	b2e5ff43...	53718853...
ef116a09...	b2e5ff43...	5df90ee1...
ef116a09...	6f5681cf...	5df90ee1...
ef116a09...	6f5681cf...	5df90ee1...
ef116a09...	6f5681cf...	53718853...

**Result: algorithm takes  $\mathcal{O}(C)$  time instead of  $\mathcal{O}(F \cdot C)$  in practice**

# Edge cases



SHA(file1.png)	SHA(file2.css)	SHA(file3.js)	SHA(file1.png)	SHA(file2.css)	SHA(file3.js)
997b0bf3...	9d055618...	5df90ee1...	997b0bf3...	9d055618...	5df90ee1...
997b0bf3...	b2e5ff43...	5df90ee1...	997b0bf3...	9d055618...	5df90ee1...
997b0bf3...	b2e5ff43...	53718853...	997b0bf3...	9d055618...	5df90ee1...
acf04ef6...	b2e5ff43...	5df90ee1...	acf04ef6...	9d055618...	5df90ee1...
ef116a09...	b2e5ff43...	53718853...	ef116a09...	6f5681cf...	53718853...
ef116a09...	6f5681cf...	5df90ee1...	ef116a09...	6f5681cf...	5df90ee1...
ef116a09...	6f5681cf...	5df90ee1...	ef116a09...	6f5681cf...	5df90ee1...
ef116a09...	6f5681cf...	5df90ee1...	ef116a09...	6f5681cf...	5df90ee1...
ef116a09...	6f5681cf...	53718853...	ef116a09...	6f5681cf...	5df90ee1...

**Needs more field experience – mine and yours as well!**

# Caching



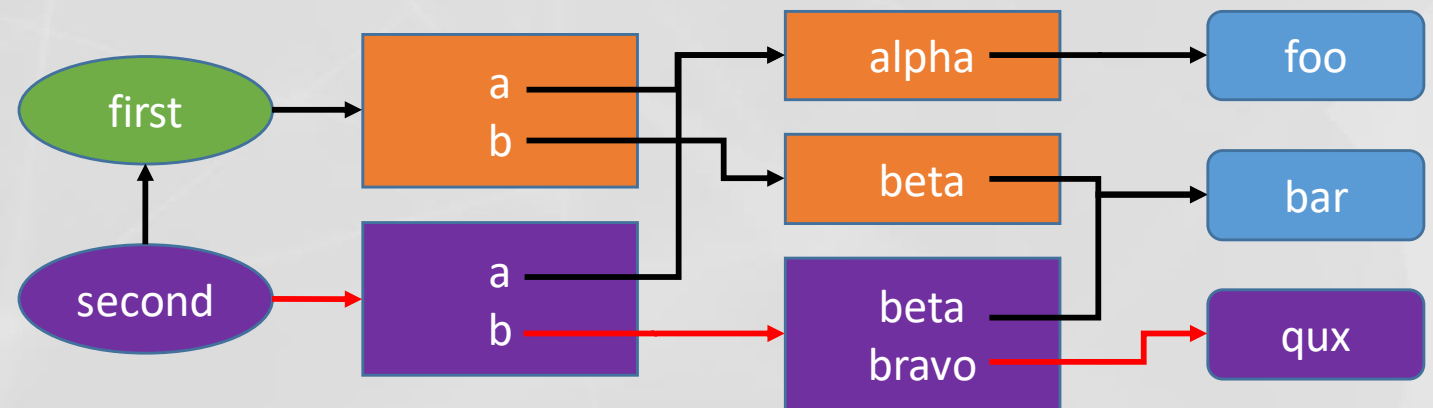
## Categories: trees that directly or indirectly...

- ...contain the blob we're looking for ( $B$ )
- ...don't contain the blob we're looking for ( $N$ )

## Observations ( $T, T'$ are trees)

- $\exists T' \in T: T' \in B \Rightarrow T \in B$
- $\forall T' \in T: T' \in N \Rightarrow T \in N$
- $|B| \ll |N|$

**TODO: persistent caching**



# SOURCE CODE



## Everything is available on GitHub

- MIT licensed
- Pull requests welcome!
- <https://github.com/silentsignal/burp-git-version>



# THANK YOU!

**ANDRÁS VERES-SZENTKIRÁLYI**

**VSZA@SILENTSIGNAL.HU**



**FACEBOOK.COM/SILENTSIGNAL**



**@SilentSignalHU**



**@dn3t**

