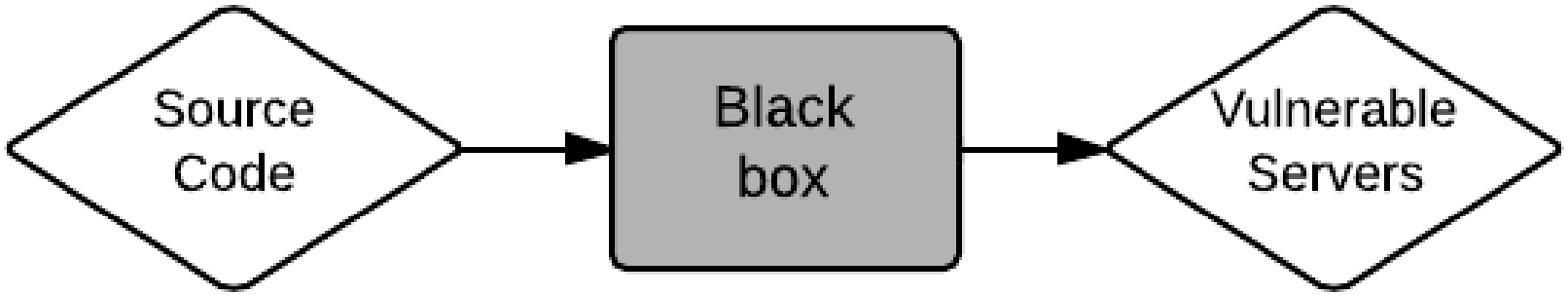
Automated vulnerability scanning and exploitation

Dennis Pellikaan Thijs Houtenbos

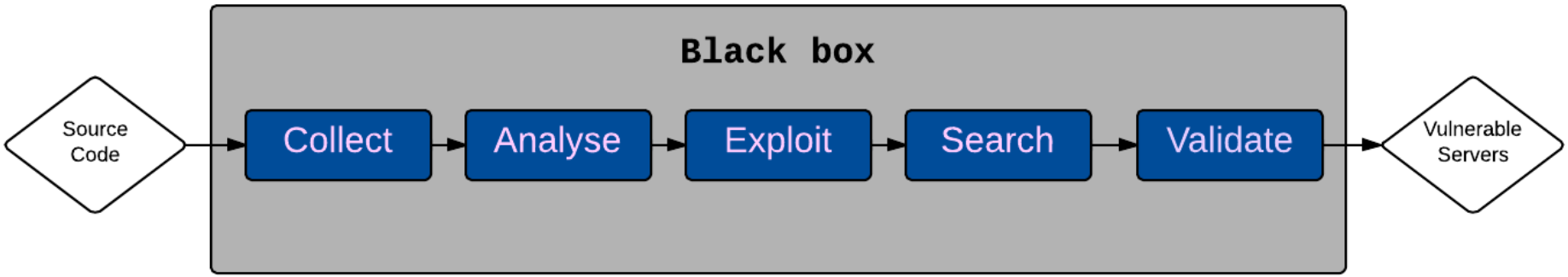
University of Amsterdam
System and Network Engineering

July 4, 2013

Introduction



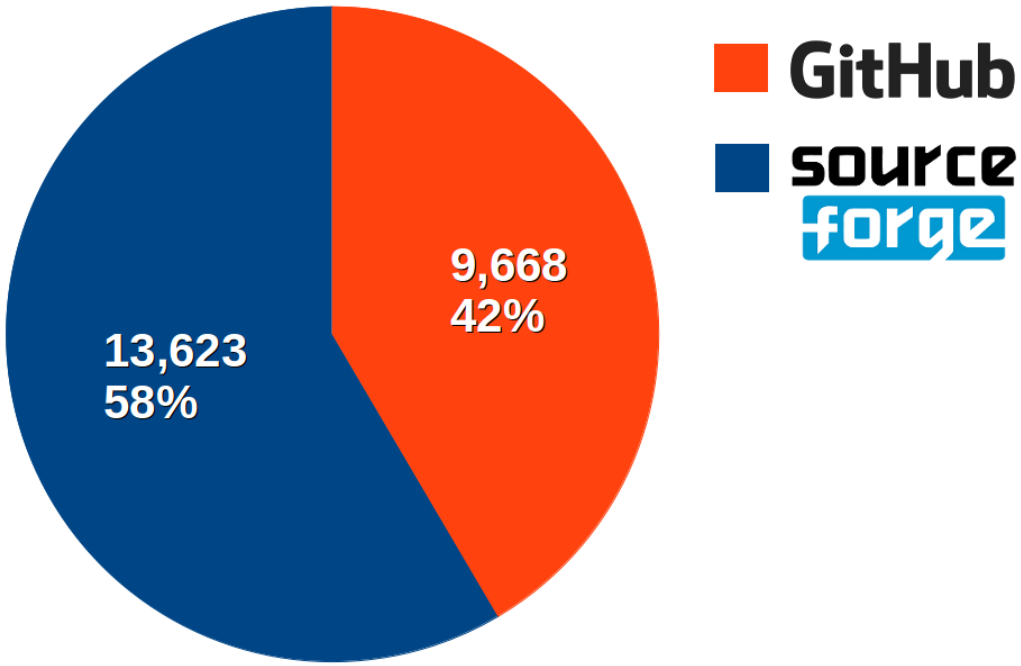
Research question



How feasible is an automated approach to compromise servers using a known source code attack on a large scale?

Collect scripts

Collected scripts



Analyse scripts

SQL Injection

mysql_query ("SELECT * FROM users WHERE id='$_GET[id]'");

File Inclusion

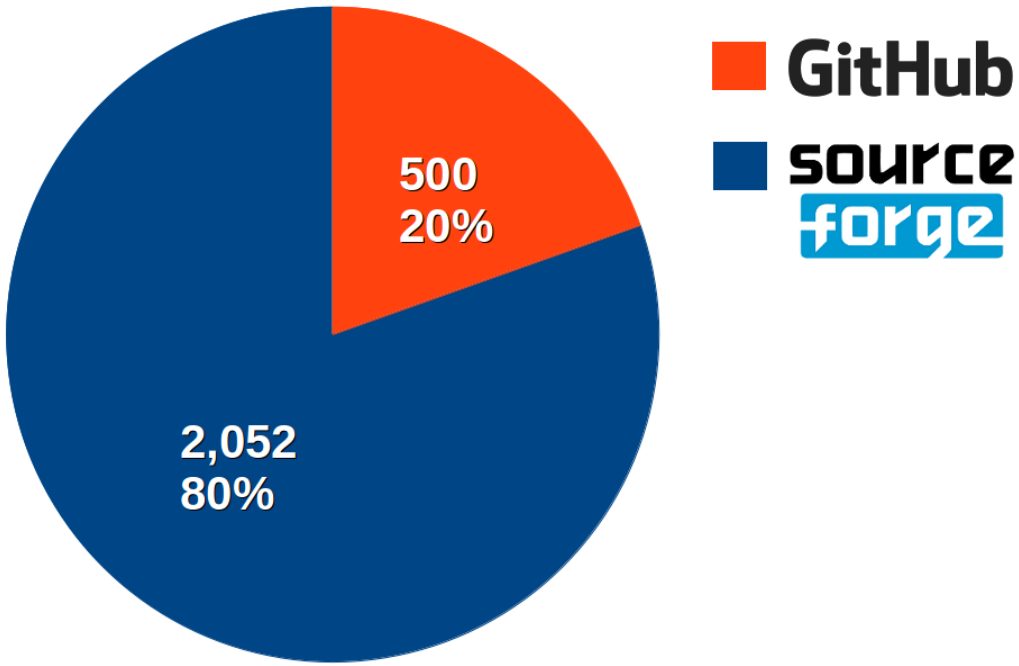
require $_POST["lang_install"].".php";

Command Injection

exec ($_GET['com'], $result);

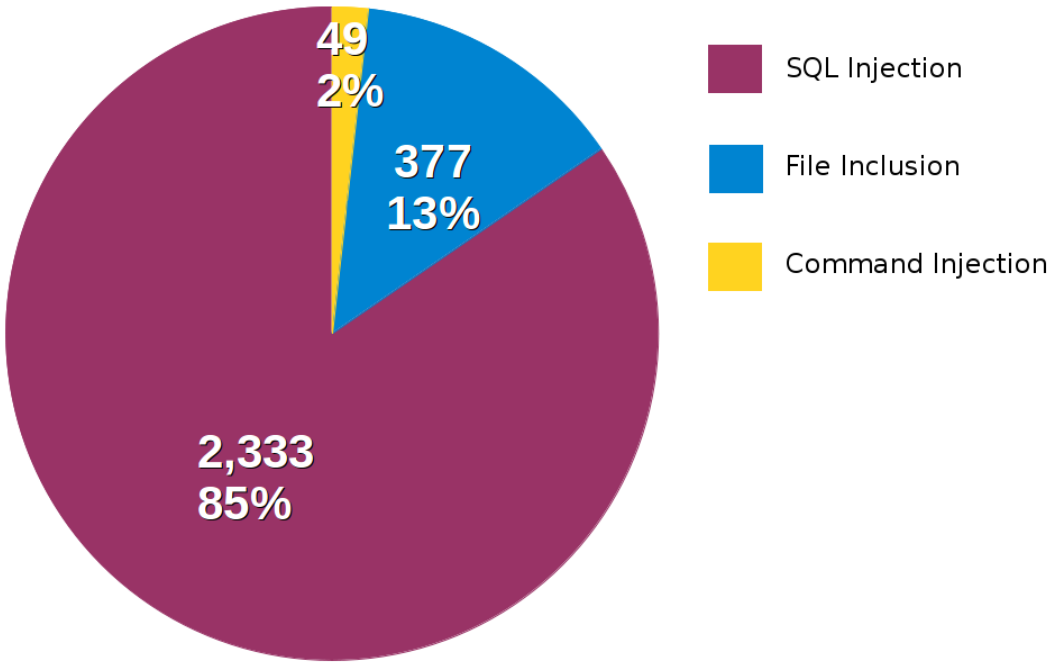
Analyse scripts

Vulnerable scripts



Analyse scripts

Vulnerable categories



Exploit vulnerabilities

SQL Injection

mysql_query ("SELECT * FROM users WHERE id='$_GET[id]'");

File Inclusion

require $_POST["lang_install"].".php";

Command Injection

exec ($_GET['com'], $result);

Exploit vulnerabilities

SQL Injection

override_function (mysql_query, log_function);

File Inclusion

338 ~~require $_POST["lang_install"].".php";~~

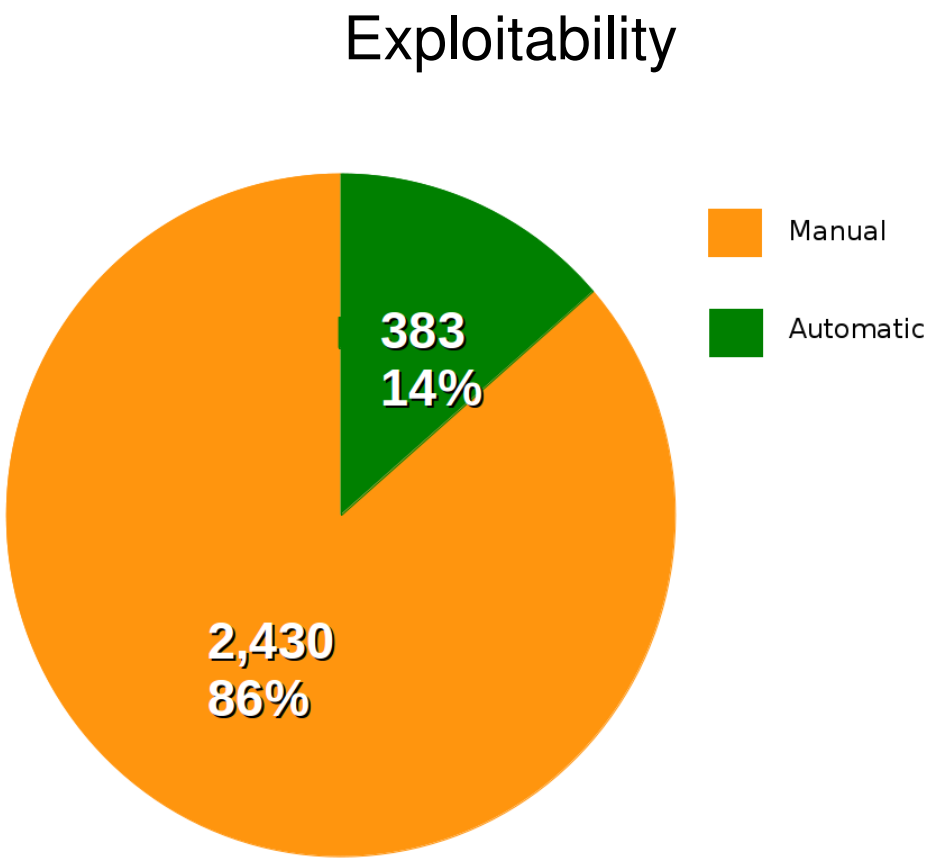
338 log_function ($_POST["lang_install"].".php");

Command Injection

183 ~~exec ($_GET['com'], $result);~~

183 log_function ($_GET['com'], $result);

Exploit vulnerabilities



Search

- Google Advanced Search Operators
  - *allinurl:"/page.php?page_id="*
  - *allintitle:"My special script v0.2a"*
- Selective results
- Rate-limiting, CAPTCHA, IPv6
- 20,000 search queries per day
- 120,000 results with 22,000 queries

Search

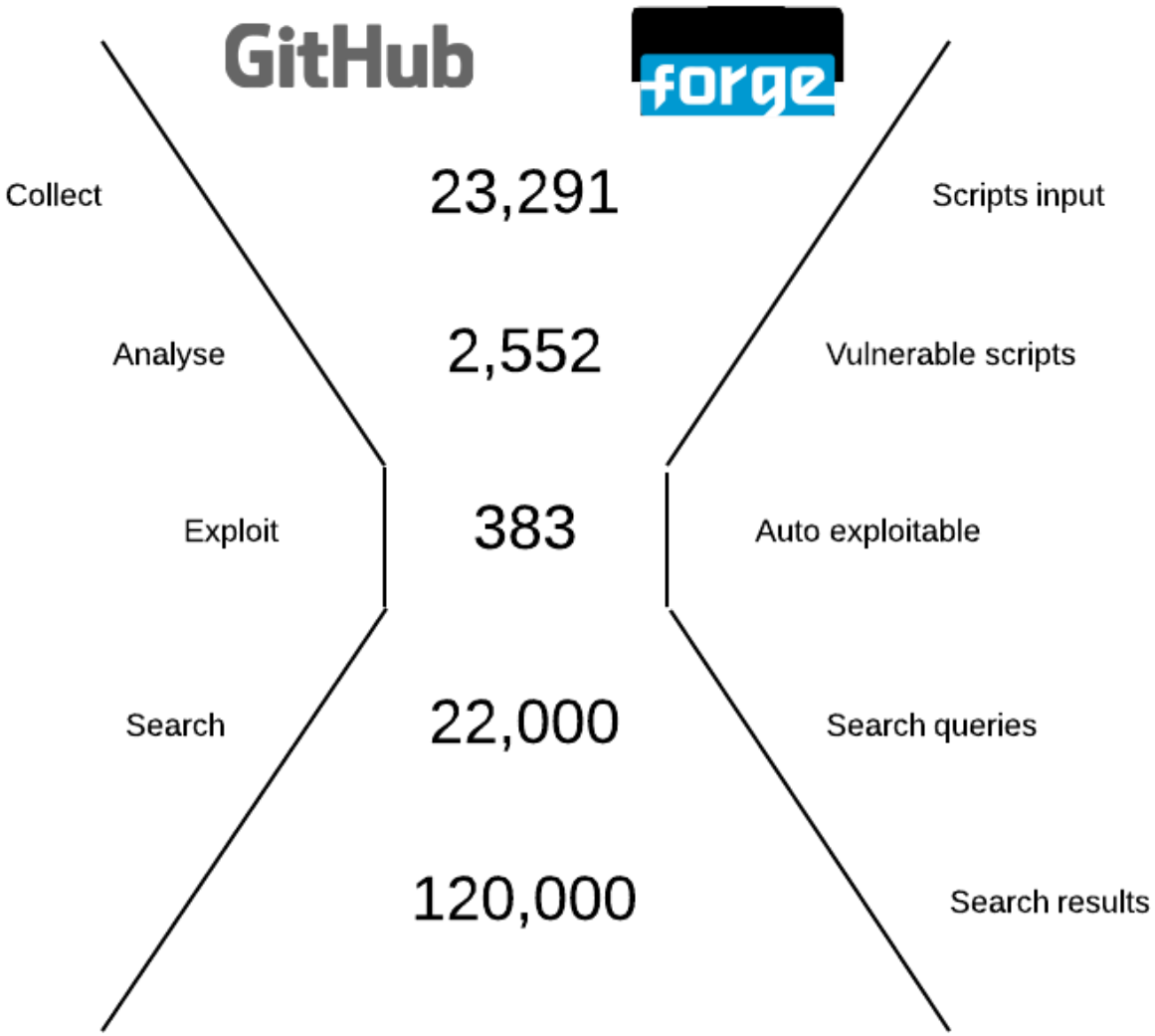

Source Code
Collect
Analyse
Exploit
Search
Validate
Vulnerable Servers


Google
allinurl:"/login.php?token="
Ongeveer 545.000 resultaten (0,15 seconden)
BZFlag Image Uploader :: Login
images.bzflag.org/submitimages/login.php?token... Vertaal deze pagina
The login was not successful. Please try again. If problem persists, please contact an administrator. By using the BZFlag Image Uploader, you agree to the Terms ...
Gooooooogle
1 2 3 4 5 6 7
Volgende

Validate search results

- Installation root
  - *http://www.example.com/users/script/install/admin.php*
  - */sourceforge/special1.0/install/admin.php*
- File comparison with bundled files (readme.txt, style.css, etc)
- Hash and text matching
- Scoring system based on matching
- 1,555 results had a perfect match
- 4,214 results had a partial match

Results


GitHub
forge
Collect
23,291
Scripts input
Analyse
2,552
Vulnerable scripts
Exploit
383
Auto exploitable
Search
22,000
Search queries
120,000
Search results

Example (1)

42 $sql = mysql_query("UPDATE users SET userid='$_GET[userid]'

Example (1)

```
42 $sql = mysql_query("UPDATE users SET userid='$_GET[userid]'
```

Example (2)

```
47 $sql="update staff set first_name='$_POST[fname]', last_name='$_POST[lname]',
middle_name='$_POST[mname]', username='".$_SESSION['admin_name']."',
password='".$_SESSION['admin_pwd']."', profile_id=1 where username='admin' ";
48 $result = mysql_query($sql);
```

Example (2)

```
47 $sql="update staff set first_name='$_POST[fname]', last_name='$_POST[lname]',
middle_name='$_POST[mname]', username='".$_SESSION['admin_name']."',
password='".$_SESSION['admin_pwd']."', profile_id=1 where username='admin' ";
48 $result = mysql_query($sql);
```

Conclusion

How feasible is an automated approach to compromise servers using a known source code attack on a large scale?

- Lots of components in the system, all with own quirks
- Almost **6,000** vulnerable servers identified
- Process can run continuously for more results
- More input is more output :-)

Questions