Ramping up Security at an open-source startup

Lukas Reschke
whois lukas@cloud.wtf
Brunei needs cyber security personnel

CYBER threats and cyber security were the main topics of discussion this week in Las Vegas at Black Hat and Def Con, two of the world's largest gatherings for security professionals and hackers. Dan Geer, who is a former chief information security officer for the NSA, said, "We need to address the cyber shortfall in the US, and that means we need to get more people into the field."
Contributor since 2012
Employed since 2014
Not much other hobbies
Fixed a lot of stuff
The good and bad of the cloud
Awesomeness of the cloud

- Accessible everywhere
- Back up online
- Easy sharing and collaboration
- All free!!!
  (or super cheap ... at least the licenses)
There is NO CLOUD, just other people’s computers

fsfe.org
Take back your data
Introducing ownCloud

- Sync and share
- Open Source
- Easy to use
- Easy to install
- Easy to extend
- > 8 million users
Web / Desktop / Mobile
But that’s not how we started...
… the project grew
... and companies started to use it
... market leader in education + research
Security at the start

- Everybody could push directly
- No formal code review process
- No static source code analysis
- No manual security testing
- No dedicated security personnel

(i.e. the same as still today in many companies ;-))
Ensuring ownCloud Security

- Pull Request reviews
Validate storage backend and auth mechanism before mounting #19313

Xenopathic commented 5 hours ago

When an admin disables a backend for user mounting, any already created personal storages should no longer be mounted for the user.

Fixes #19312

cc @PVince81 @DeepDiver1975 @icewind1991

Xenopathic added some commits 5 hours ago

- Perform visibility checks on storages ...
- Unit tests for storage validation in service

Xenopathic added bug 3 - To Review app:files_external regression labels 5 hours ago

Xenopathic added this to the 8.2-current milestone 5 hours ago

PVince81 commented 4 hours ago

👍 works

I also tested regiving the permission and the storages were back.
"Ask programmers to review 10 lines of code, they'll find 10 issues. Ask them to do 500 lines and they'll say it looks good".

The harsh truth about code reviews.

Source: https://twitter.com/paulmgower/status/67441209836351488
Ensuring ownCloud Security

• Pull Request reviews
• Regular code reviews for security issues
As a result of the security check conducted in September multiple critical vulnerabilities have been found, including:

1. Arbitrary local file disclosure via the office document conversion component. (similar to the issue found in the previews module)
2. Multiple XSS vulnerabilities
3. The documents application allows unauthenticated user to edit any public shared file
4. Bypass of the password protection
5. Allows logged-in users to access files of other users
6. A JSON injection

Furthermore, the SQL query in `op.php` is vulnerable against SQL injections if called with user-supplied parameters:

```php
protected function hasOp($esId, $memberId, $opType){
    $ops = $this->execute(
        'SELECT * FROM `'. $this->tableName . '` WHERE `es_id`=? AND `opspec` LIKE array($esId)
    )
    $result = $ops->fetchAll();
    return is_array($result) && count($result)>0;
}
```

### Acceptance criteria

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Risk</th>
<th>Status</th>
<th>Verified</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>The application <strong>MUST</strong> verify whether the user has all required privileges to access or modify the file.</td>
<td>High</td>
<td>✗</td>
<td>7.0.2</td>
<td>1</td>
</tr>
<tr>
<td>The application <strong>MUST NOT</strong> leak any local files</td>
<td>High</td>
<td>~</td>
<td>7.0.2</td>
<td>2</td>
</tr>
</tbody>
</table>
Ensuring ownCloud Security

- Pull Request reviews
- Regular code reviews for security issues
- Automated static analysis
public function fixEOL($str)
{
    // Normalise to \n
    $nstr = str_replace(array("\r\n", "\r"), "\n", $str);

    // Now convert LE as needed
    if ($this->LE !== "\n") {
        $nstr = str_replace("\n", $this->LE, $nstr);
    }

    return $nstr;

} /* Add a custom header. */
/* $name value can be overloaded to contain */
Ensuring ownCloud Security

• Pull Request reviews
• Regular code reviews for security issues
• Automated static analysis
• Customers do perform security tests
• Following industry best practice for security handling (oriented towards ISO 29147, 30111 and 27304)
Security Advisory

Title

Risk

Common Weakness Enumeration

Vulnerability description

Affected software + patches + CVE

What we did to fix it

Credits

---

PHP arbitrary class instantiation in "files_external" (oC-SA-2015-018)

30th September 2015

Risk level: High

CVSS v2 Base Score: 9 (AV:N/AC:L/Au:S/C:C/I:C/A:C)

CWE: Use of Externally-Controlled Input to Select Classes or Code ('Unsafe Reflection') (CWE-470)

Description

A user may instantiate arbitrary ownCloud classes due to a lack of a proper check of the mount point options provided by a user via the web frontend. These may include constructor arguments and could potentially lead to a remote code execution.

Affected Software

- ownCloud Server < 8.1.2 (CVE-2015-7699)
  - core/a1706f61af822aeba4ea9e84b53c5cea984f8e4
- ownCloud Server < 8.0.7 (CVE-2015-7699)
  - core/595381b9db5d676492ff8957de0590982ed1864a4
- ownCloud Server < 7.0.9 (CVE-2015-7699)
  - core/b05e178bf884b120d1106e6a28f35aa50d6d6f

Action Taken

The mount points are now properly validated in the controller before being stored.

Acknowledgements

The ownCloud team thanks the following people for their research and responsible disclosure of the above advisory:

- Robin McCorkell - ownCloud Inc. (rmccorkell@owncloud.com) - Vulnerability discovery and disclosure.
Security work going on in ownCloud

May 26, 2015

Besides a lot of the performance work that was lately done as well as the stability and architectural improvements we work on, we are also striving to make ownCloud even more secure by improving our API as well as introducing new hardening features. In this blog post I am going to feature some of these changes.

Those include:

- "data/.htaccess" is updated after each update
- SabreDAV web interface has been removed
- Trusted domains are now a hard requirement
- Failed login log entries supports reverse proxy settings
- Request ID supports mod_unique_id
- "From link" download functionality has been removed
- Security-related headers can now be sent by the web server
- Enhancement of root certificate handling
- OC_User_HTTP backend replaced by user_webdavauth
- Random-number generator returns all base64 characters
- OC.generateUrl encodes parameters now automatically
- Code checker checks for strict comparisons in PHP code
- Constructor of OC\Files\View prevents directory traversals
- Stricter Content-Security-Policy and more choices for developers
- New security guidance and tips & tricks
And yet...

HATERS GONNA HATE
And yet...

Happy 5th birthday #owncloud. Here's 96 reasons to avoid you. cvedetails.com/vulnerability-…
How are we doing?

Internet Explorer vs Murder Rate

- Murders in US
- Internet Explorer Market Share

- 2006: 18,000
- 2007: 17,200
- 2008: 16,400
- 2009: 15,600
- 2010: 14,800
- 2011: 14,000

- 2006: 90%
- 2007: 75%
- 2008: 60%
- 2009: 45%
- 2010: 30%
How are we doing?

**Fixed vulnerabilities per year**

<table>
<thead>
<tr>
<th>Year</th>
<th>Server</th>
<th>Community Apps</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How are we doing?

Severity by reporter in ownCloud Server since 2012

- Low
- Medium
- Critical
How are we doing?

Severity by reporter in ownCloud Server in 2012

- Low
- Medium
- Critical

Legend:
- Internal
- External
How are we doing?

Severity by reporter in ownCloud Server in 2014

- **Low**: Internal (3), External (1)
- **Medium**: Internal (8), External (4)
- **Critical**: Internal (26)

Legend:
- **Blue**: Internal
- **Green**: External
How are we doing?

Severity by reporter in ownCloud Server in 2015

- Critical: Internal
- Medium: Internal
- Low: Internal and External
Security: Secure by Default!

- Security checks have to be disabled by the developer (e.g. CSRF + authentication)
Security: Secure by Default!

- Sicherheitschecks müssen von Entwicklern bewusst deaktiviert werden.

```php
/**
 * @PublicPage
 * @NoCSRFRequired
 *
 * @param string $token
 * @param string $path
 * @return TemplateResponse|RedirectResponse
 * @throws NotFoundException
 */

public function showShare($token, $path = '') {
    \OC_User::setIncognitoMode(true);

    // Check whether share exists
    $linkItem = Share::getShareByToken($token, false);
    if ($linkItem === false) {
        return new NotFoundResponse();
    }

    $shareOwner = $linkItem['uid_owner'];
    $originalSharePath = $this->getPath($token);

    // Share is password protected – check whether the user is permitted
    if (isset($linkItem['share_with']) && !Helper::authenticate($linkItem)) {
        return new RedirectResponse($this->urlGenerator->linkToRoute('filesystem',
            array('token' => $token)));
    }

    if (Filesystem::isReadable($originalSharePath . $path)) {
        $getFilesPath = Filesystem::normalizePath($path);
```
Security: Secure by Default!

• Security checks have to be disabled by the developer (e.g. CSRF + Authentication)

• Internal file system not vulnerable against directory traversal
```php
/**
 * @param string $token
 * @param string $path
 * @return TemplateResponse|RedirectResponse
 * @throws NotFoundException
 */

public function accessFolder($token, $path = '') {
    $view = new OC\Files\View('/folderToChroot/');
    // Funktioniert
    $view->fopen('myfile.txt', 'r');
    // Funktioniert NICHT
    $view->fopen('../../myfile.txt', 'r');
```
Security: Secure by Default!

- Security checks have to be disabled by the developer (e.g. CSRF + Authentication)
- Internal file system not vulnerable against directory traversal
- Security functionalities are enabled by default in ownCloud server (e.g. Content-Security-Policy)
- ...

Potential dangerous PHP functions are blacklisted

```
1. lukasreschke@Lukass-MBP: ~/Documents/Programming/master (zsh)
line 34: OC_Util - Static method of private class must not be called
5 errors
line 42: OC_Helper - Static method of private class must not be called
line 45: OCP\Response - Static method of deprecated class must not be called
line 46: OCP\Response - Static method of deprecated class must not be called
line 47: OCP\Response - Static method of deprecated class must not be called
line 49: OC_Util - Static method of private class must not be called
6 errors
line 61: OC_Util - Static method of private class must not be called
line 63: OCP\App::setActiveNavigationEntry - Method of deprecated class must not be called
line 82: OCP\Util::linkTo - Method of deprecated class must not be called
line 86: OC_User - Static method of private class must not be called
line 92: OC_Helper - Static method of private class must not be called
line 118: OC_App - Static method of private class must not be called
7 errors
line 45: OC_Helper - Static method of private class must not be called
line 46: OC_L10N - private class must not be instantiated
line 69: OC_Helper - Static method of private class must not be called
line 72: OC_Helper - Static method of private class must not be called
line 74: OC_Helper - Static method of private class must not be called
line 77: OC_Helper - Static method of private class must not be called
line 140: OCP\Util::formatDate - Method of deprecated class must not be called
1 errors
line 94: OCP\Util::linkTo - Method of deprecated class must not be called

Deprecated field available: shipped => true
Deprecated field available: standalone
Migrate the app version to appinfo/info.xml (add <version>1.1.11</version> to appinfo/info.xml and remove appinfo/version)
App is not compliant
→ master git:(master)  
```
Security is hard
No technology is perfect, and ownCloud Inc. believes that working with skilled security researchers across the globe is crucial in identifying weaknesses in any technology. We created the ownCloud Security Bug Bounty Program to reward security researchers for finding issues in the ownCloud Server, and in so doing help strengthen ownCloud Server for customers, users and the community. If the submitter chooses not to get a bounty, emailed to the ownCloud security mailing list (security@owncloud.com).

For the time being we want to focus our efforts on ownCloud Server instead of vulnerabilities within our website or infrastructure. However, if you find a security bug in our website or infrastructure we welcome any report, though we cannot offer a monetary incentive. Please do read the defined scope below carefully.

DO NOT RUN AUTOMATED SCANNERS AGAINST *.OWNCLOUD.COM AND *.OWNCLOUD.ORG SERVERS. AUTOMATED SCANS AGAINST OUR INFRASTRUCTURE MAY BE MARKED AS NOT APPLICABLE.

If you have questions please direct them to lukas@owncloud.com

Rewards

If the bug is identified as meaningful and qualifies for the program, and the submitter has followed the Disclosure Policy, the bug bounty is paid out on the following schedule by bug severity:

<table>
<thead>
<tr>
<th>Impact</th>
<th>Definition</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>Critical</td>
<td>Giving the adversary complete control over the server. (RCE / SQL injection / ...)</td>
<td>$250-$500</td>
</tr>
</tbody>
</table>

$25
Minimum bounty

20
Hackers thanked

21
Reports closed

Top Hackers

ishahriyar
Reputation: 37

ashesh
Reputation: 30

psych01r1a
Reputation: 14

mandeep
Reputation: 9

00day
Reputation: 7
Why HackerOne?

• Used by other major vendors

• Great triaging tools and support
• Payments processed by HackerOne
The platform

SMTP Protection not used

#89856

Reporter: ahmedadel

24 hours ago

I'm checking your website found spf record there. You should apply strict SMPT policy to stop spoofed email sending from your domain. An attacker would send a fake email from info@owncloud.com saying that please change your password, the victim is aware of phishing attacks, but when he sees that the mail originated from info@owncloud.com, he has no other way than to believe it. Clicking on the link takes him to a website where certain JavaScript is executed which steals his owncloud.com id and password (SESSION COOKIE). The results can be more dangerous.

When I tried to send an email from info@owncloud.com to my email, it was successful but when I tried to send another from admin@facebook.com or any other, I did not receive any email. Hence, there might be some configuration missing in your mail servers.

Fix:

Your SPF record is "v=spf1 mx include:_spf.google.com a:kerio.owncloud.com a:mail.hive01.com include:email1.com include:email.infituitive.com ~all".

It should be "v=spf1 mx includes:_spf.google.com a:kerio.owncloud.com a:mail.hive01.com include:email1.com include:email.infituitive.com ~all".
The platform

HackerOne, Inc. [US] https://hackerone.com/bugs?subject=owncloud&report_id=89856&view=closed&substates%5B%5D=resolved&...

**Open (26)** **Closed (225)**

Search all reports

---

**hackbot** posted an internal comment.

Hey there! I am **Hackbot**, I help find possible duplicates and related reports. Here are my top suggestions:

- **(82%) Report #83253** (informative): SPF Protection not used, I can hijack your email server

I also found some public reports from other teams that are similar to this report:

- **(78%) Report #77060** (resolved): SMTP protection not used
- **(77%) Report #56177** (resolved): SMTP protection not used
- **(67%) Report #25191** (informative): SMTP protection not used (please read carefully)
- **(51%) Report #57736** (resolved): Missing spf flags for hackerone.com
- **(50%) Report #34112** (resolved): SMPT Protection not used, I can hijack your email server.

---

**lukasreschke** closed the bug and changed the status to Duplicate (#83253).

Thanks a lot for reporting this issue back to us. We do however have employed a policy using ~all on purpose since we’re running mailing lists as well as other software on the same domain.

While we might consider to adjust this in the future for now we do consider this an acceptable behaviour and I will close this issue. See http://serverfault.com/questions/611575/will-mailing-lists-break-if-spf-is-too-restrictive for some considerations on problems with mailing lists and a too restrictive SPF policy.
New Reports

Invited top 600 HackerOne users
(... each of them got a mail...)

Public launch
... and?

**Resolved reports**
- 3 bugs in scope ($700)
- 43 bugs out of scope

**Type of reported bugs**
- Duplicate: 36%  
- Informative: 32%  
- Not applicable: 18%  
- Resolved: 14%
Lessons learned from a bug bounty program

• Protect infrastructure against automated testing tools in advance
  – Don’t forget the contacts form

• Quality of reports differs hugely depending on the reporter

• Likely no low hanging fruits
What went wrong?
What could have been better?
Pull Request Reviews

• Added at a late stage.
• Prevents a lot of pitfalls
• ... ensure they actually get reviewed ...
## OWNCLOUD 4.0 AND ENCRYPTION

Recently, the new version 4.0 of the OwnCloud open-source software has been released. According to Wikipedia, "OwnCloud is a software suite that provides a location-independent storage area for data (cloud storage). The project was launched in January 2010 from KIGI developer Frank Karlitschek to create a free alternative to commercial cloud providers. In contrast to commercial storage services, ownCloud can be installed on a private server at no additional cost. So, anybody sensitive to the privacy of his own data, but still willing to own them in the cloud, might be tempted to install the feature-rich OwnCloud application on a dedicated server.

Even more interestingly, the feature list of the latest version mentions the following:

- Do you want to make sure that your files remain secure on the server? With the Encryption Application enabled, all files stored on the ownCloud server are encrypted to your password so not even the admin can look inside your files. Add to this an SSL connection, and your files are secure while in motion and at rest.

I did not resist to take a closer look at these claims.

The file-encryption capabilities are implemented as the OC_Crypt class. Here is a quick summary of my findings:

- The key is generated using four calls to the mt_rand() PHP routine, which implements the Mersenne Twister pseudo-random generator and is unfortunately not of cryptographic quality:

```plaintext
public static function createKey($username, $password) {
    // generate 16 random bytes
    $sk = mt_rand(0x0000, 0xFFFF);
    $sk = mt_rand($sk, $sk) + mt_rand(0x0000, 0xFFFF);
    // encrypt the key with the password of the user
    $encKey = OC_Crypt::encrypt($sk, $password);
    // Write the file
    $encryptedKey = OC_FileProxy::encode($encKey);
    $file = OC_Move::create('core_downloads/encryption.key', $encryptedKey);
    if ($file) {
        // Permissions
        $filesystem = OC_Filesystem::getSystem();
        $filesystem->setProperty($username, '/encryption.key', $sk);
        $fileProxy = OC_FileProxy::instance($file);
        $fileProxy->setPermission('readable', 'group', 0644);
    } else {
        // Error
    }
}
```

Openness

• In retrospective: Consider publishing advisories first after you consider your project secure enough.
External reviews

- Reviews will come anyways.
- Best to be pro-active and have stuff fixed before.
- Bug bounty a good addition to external reviews.
  - ... consider starting with higher rewards though.

Do not trust reviews without checking them in detail.
Don’t fix single bugs

... fix the categories of bugs and do root cause analysis.
Thanks!
github.com/owncloud
hackerone.com/owncloud