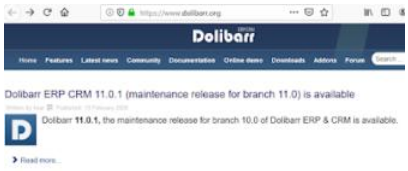
SOBOTA, 15 LUTEGO 2020

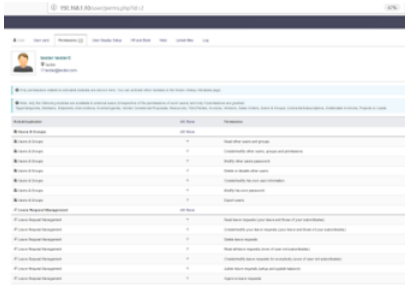
Exploiting Dolibarr 11

This time I tried to check one of the ERP/CRM software available on the market. I decided to try latest version of Dolibarr from Bitnami resources (. Below you will find few notes about it. Here we go...

This time we will start here:

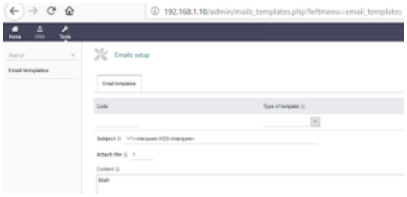


When your VM is ready, login in as an admin (in case of Bitnami VMs it will probably be 'user'). There you will will be able to create new users. I started from user 'tester' with 'no permissions':



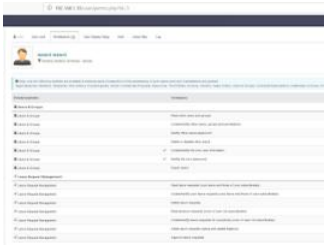
Let's try to find if there are any bugs when (1st 'normal') user ('tester') is logged-in:

As you can see (in *default* Bitnami's installation) for 'registered user with no perms' there is only one available link to use - 'Email templates'. Let's try it:



Quick results:

Next I created new user: tester2. This time I added few permissions, see below:



Ok, our user 'can do' something now. ;) Let's try to *personalize* our profile a little bit:

O MNIE

code16

Cody Sixteen

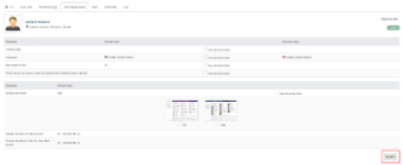
Wyświetl mój pełny profil

ARCHIWUM BLOGA

► 2022 (16)
► 2021 (37)
▼ 2020 (62)
► 12 (1)
► 11 (2)
► 10 (1)
► 09 (2)
► 08 (5)
► 07 (5)
► 06 (7)
► 05 (5)
► 04 (11)
► 03 (10)
▼ 02 (6)
Postauth RCE in Centreon 19.10 - part 2
Postauth RCE in Centreon 19.10
Bug bounty scam program
Exploiting Dolibarr 11
Escaping from the Fort - quick CVE-2017-14187 autopsy
Trying harder...
► 01 (7)
► 2019 (97)
► 2018 (67)
► 2017 (58)
► 2016 (63)

ETYKIETY

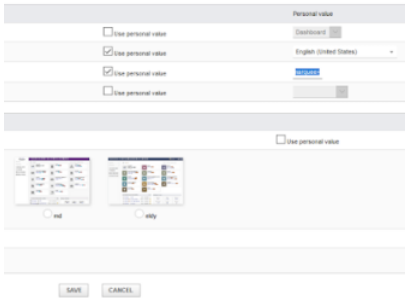
.net
android
binary
crackme
ctf
debug
docker
drones
enlil
FortiGate
fuzz
infrastructure
malware
notes
pentest
poc
pwn
RE
web
writeup



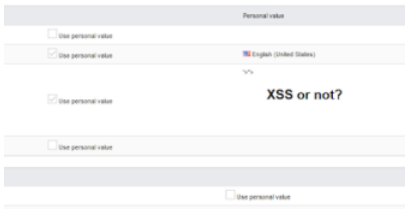
For this one injection I used similar code:

```
' > " > < h 1 > < marquee >XSS or not?< h1 > < / marquee >
```

(just remove ' ' ;))



Checking:

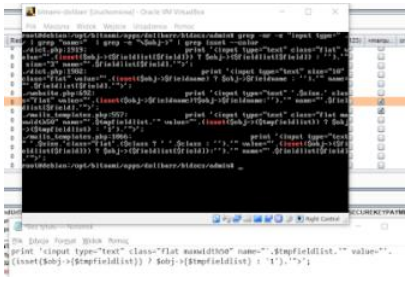


I believe it's *personalised* now. ;]

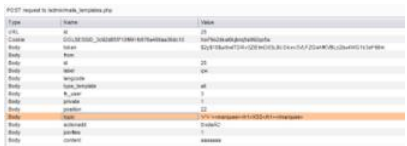
Next (checking response in Burp):

I was wondering if I can achieve similar results - simply checking the source for the same *pattern ;)*

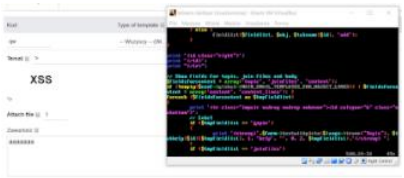
Checking:



As you can see (*grep* above) we should be able to get those bugs from 'admin user logged-in', so it's time to switch. Checking:



Response:



Cool. Next one:

Results:

So for now vulnerable parameters are: *joinfiles, topic, code.*

In case you're looking for nice *Referer* - this one should be good:

Response in Burp:

Response in the browser:

Yep. :)

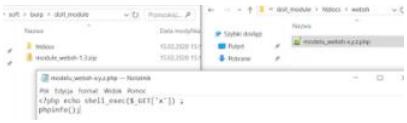
But when we're talking about the admin-access we should mention about one nice thing - modules. :)

I believe you're already pretty familiar with what will happen next ;) - so here we go:

As you can see 'admin user' is able to upload/add *new module*. To do that (with Bitnami) you'll need to add *write* perms to the location mentioned in the screen below:

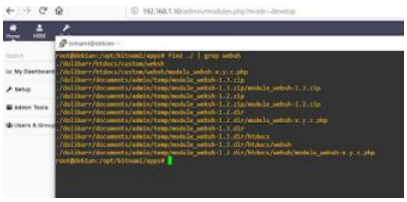
When it's done - we can continue. Next:

Preparing our module:



Next:

Last thing to find is the location of our webshell:



Ok, that should be easy:

I think that's all. ;)

See you next time!

Cheers

Posted by code16 at 14:20

Labels: napalm, notes, pentest, poc, web, writeup

Brak komentarzy:

Prześlij komentarz

Wpisz komentarz