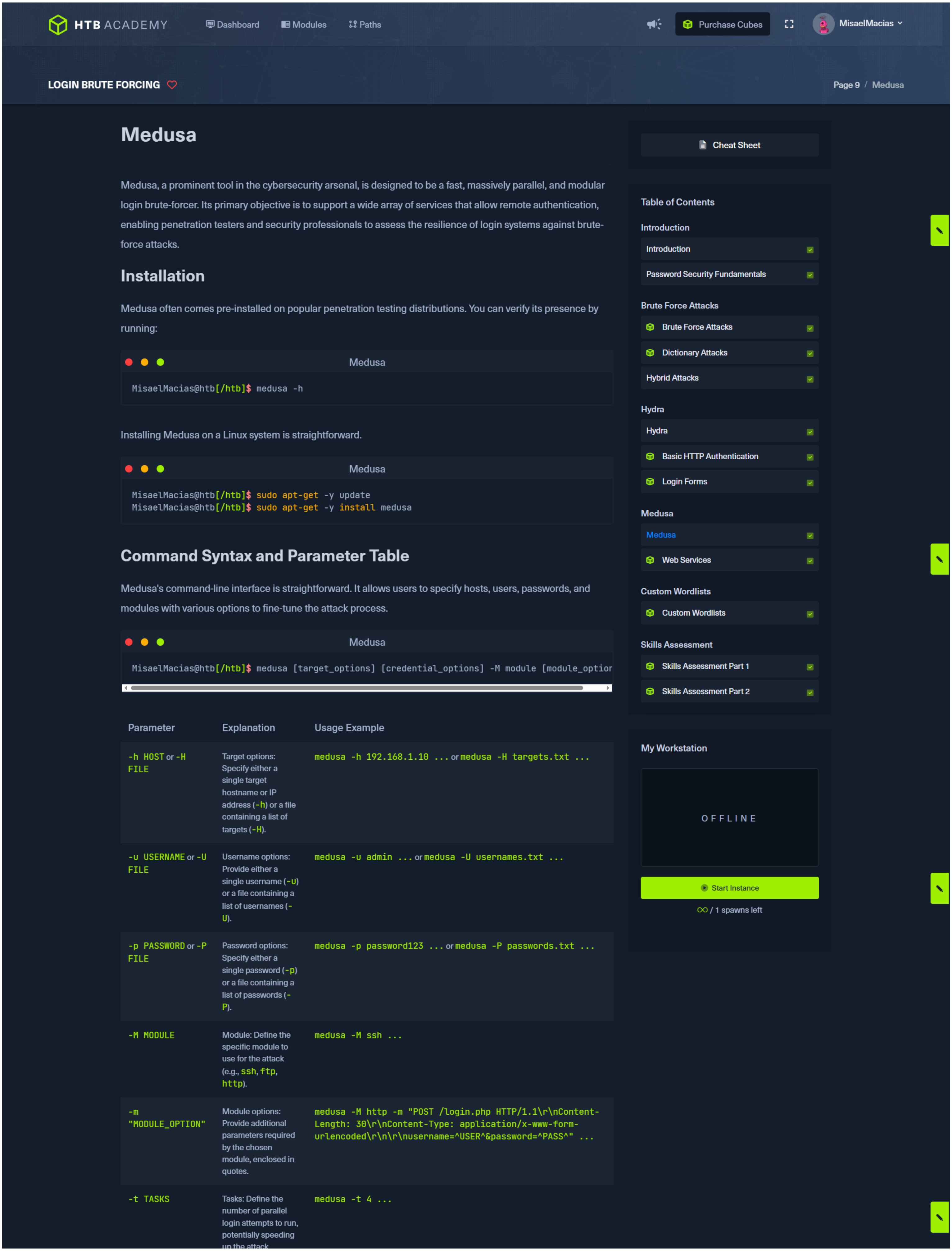

Medusa

Medusa, a prominent tool in the cybersecurity arsenal, is designed to be a fast, massively parallel, and modular login brute-forcer. Its primary objective is to support a wide array of services that allow remote authentication, enabling penetration testers and security professionals to assess the resilience of login systems against brute-force attacks.

Installation

Medusa often comes pre-installed on popular penetration testing distributions. You can verify its presence by running:

Medusa

```
MisaelMacias@htb[/htb]$ medusa -h
```

Installing Medusa on a Linux system is straightforward.

Medusa

```
MisaelMacias@htb[/htb]$ sudo apt-get -y update
MisaelMacias@htb[/htb]$ sudo apt-get -y install medusa
```

Command Syntax and Parameter Table

Medusa's command-line interface is straightforward. It allows users to specify hosts, users, passwords, and modules with various options to fine-tune the attack process.

Medusa

```
MisaelMacias@htb[/htb]$ medusa [target_options] [credential_options] -M module [module_optior
```

Parameter	Explanation	Usage Example
`-h HOST` or `-H FILE`	Target options: Specify either a single target hostname or IP address (`-h`) or a file containing a list of targets (`-H`).	`medusa -h 192.168.1.10 ...` or `medusa -H targets.txt ...`
`-u USERNAME` or `-U FILE`	Username options: Provide either a single username (`-u`) or a file containing a list of usernames (`-U`).	`medusa -u admin ...` or `medusa -U usernames.txt ...`
`-p PASSWORD` or `-P FILE`	Password options: Specify either a single password (`-p`) or a file containing a list of passwords (`-P`).	`medusa -p password123 ...` or `medusa -P passwords.txt ...`
`-M MODULE`	Module: Define the specific module to use for the attack (e.g., `ssh`, `ftp`, `http`).	`medusa -M ssh ...`
`-m "MODULE_OPTION"`	Module options: Provide additional parameters required by the chosen module, enclosed in quotes.	`medusa -M http -m "POST /login.php HTTP/1.1\r\nContent-Length: 30\r\nContent-Type: application/x-www-form-urlencoded\r\n\r\nusername=^USER^&password=^PASS^" ...`
`-t TASKS`	Tasks: Define the number of parallel login attempts to run, potentially speeding up the attack	`medusa -t 4 ...`

Cheat Sheet

Table of Contents

My Workstation

OFFLINE

Start Instance

∞ / 1 spawns left

	up the attack.	
`-f` or `-F`	Fast mode: Stop the attack after the first successful login is found, either on the current host (`-f`) or any host (`-F`).	`medusa -f ...` or `medusa -F ...`
`-n PORT`	Port: Specify a non-default port for the target service.	`medusa -n 2222 ...`
`-v LEVEL`	Verbose output: Display detailed information about the attack's progress. The higher the `LEVEL` (up to 6), the more verbose the output.	`medusa -v 4 ...`

Medusa Modules

Each module in Medusa is tailored to interact with specific authentication mechanisms, allowing it to send the appropriate requests and interpret responses for successful attacks. Below is a table of commonly used modules:

Medusa Module	Service/Protocol	Description	Usage Example
FTP	File Transfer Protocol	Brute-forcing FTP login credentials, used for file transfers over a network.	`medusa -M ftp -h 192.168.1.100 -u admin -P passwords.txt`
HTTP	Hypertext Transfer Protocol	Brute-forcing login forms on web applications over HTTP (GET/POST).	`medusa -M http -h www.example.com -U users.txt -P passwords.txt -m DIR:/login.php -m FORM:username=^USER^&password=^PASS^`
IMAP	Internet Message Access Protocol	Brute-forcing IMAP logins, often used to access email servers.	`medusa -M imap -h mail.example.com -U users.txt -P passwords.txt`
MySQL	MySQL Database	Brute-forcing MySQL database credentials, commonly used for web applications and databases.	`medusa -M mysql -h 192.168.1.100 -u root -P passwords.txt`
POP3	Post Office Protocol 3	Brute-forcing POP3 logins, typically used to retrieve emails from a mail server.	`medusa -M pop3 -h mail.example.com -U users.txt -P passwords.txt`
RDP	Remote Desktop Protocol	Brute-forcing RDP logins, commonly used for remote desktop access to Windows systems.	`medusa -M rdp -h 192.168.1.100 -u admin -P passwords.txt`
SSHv2	Secure Shell (SSH)	Brute-forcing SSH logins, commonly used for secure remote access.	`medusa -M ssh -h 192.168.1.100 -u root -P passwords.txt`
Subversion (SVN)	Version Control System	Brute-forcing Subversion (SVN) repositories for version control.	`medusa -M svn -h 192.168.1.100 -u admin -P passwords.txt`
Telnet	Telnet Protocol	Brute-forcing Telnet services for remote command execution on older systems.	`medusa -M telnet -h 192.168.1.100 -u admin -P passwords.txt`
VNC	Virtual Network Computing	Brute-forcing VNC login	`medusa -M vnc -h 192.168.1.100 -P passwords.txt`

		credentials for remote desktop access.	
Web Form	Brute-forcing Web Login Forms	Brute-forcing login forms on websites using HTTP POST requests.	`medusa -M web-form -h www.example.com -U users.txt -P passwords.txt -m FORM:"username=^USER^&password=^PASS^:F=Invalid"`

Targeting an SSH Server

Imagine a scenario where you need to test the security of an SSH server at `192.168.0.100`. You have a list of potential usernames in `usernames.txt` and common passwords in `passwords.txt`. To launch a brute-force attack against the SSH service on this server, use the following Medusa command:

Medusa

```
MisaelMacias@htb[/htb]$ medusa -h 192.168.0.100 -U usernames.txt -P passwords.txt -M ssh
```

This command instructs Medusa to:

- Target the host at `192.168.0.100`.
- Use the usernames from the `usernames.txt` file.
- Test the passwords listed in the `passwords.txt` file.
- Employ the `ssh` module for the attack.

Medusa will systematically try each username-password combination against the SSH service to attempt to gain unauthorized access.

Targeting Multiple Web Servers with Basic HTTP Authentication

Suppose you have a list of web servers that use basic HTTP authentication. These servers' addresses are stored in `web_servers.txt`, and you also have lists of common usernames and passwords in `usernames.txt` and `passwords.txt`, respectively. To test these servers concurrently, execute:

Medusa

```
MisaelMacias@htb[/htb]$ medusa -H web_servers.txt -U usernames.txt -P passwords.txt -M http -
```

In this case, Medusa will:

- Iterate through the list of web servers in `web_servers.txt`.
- Use the usernames and passwords provided.
- Employ the `http` module with the `GET` method to attempt logins.

By running multiple threads, Medusa efficiently checks each server for weak credentials.

Testing for Empty or Default Passwords

If you want to assess whether any accounts on a specific host (`10.0.0.5`) have empty or default passwords (where the password matches the username), you can use:

Medusa

```
MisaelMacias@htb[/htb]$ medusa -h 10.0.0.5 -U usernames.txt -e ns -M service_name
```

This command instructs Medusa to:

- Target the host at `10.0.0.5`.
- Use the usernames from `usernames.txt`.
- Perform additional checks for empty passwords (`-e n`) and passwords matching the username (`-e s`).
- Use the appropriate service module (replace `service_name` with the correct module name).

Medusa will try each username with an empty password and then with the password matching the username, potentially revealing accounts with weak or default configurations.

← Previous   Next →   ✓ Mark Complete & Next