

# Day 20 - Mythic Server Setup

# Mythic Server Setup

Procediamo col deployment di **C2 Server** con le seguenti caratteristiche:



## Cloud Compute - Shared CPU

Virtual machines for apps with bursty performance, e.g. low traffic websites, blogs, CMS, dev/test environments, and small databases.



Toronto

Canada



Ubuntu

22.04 LTS x64

Server Hostname

Enter server hostname (6/63)

Mythid

Server Label

Enter server label

Mythic



80 GB SSD

2 vCPUs

4 GB

80 GB SSD

3 TB

\$20/month

\$0.027/hour

# Mythic Server Setup

Utilizzeremo anche una macchina virtuale **Kali Linux** già esistente su **VirtualBox**, con la scheda di rete impostata su **modalità bridge** (bridged mode).

**Rete**

Scheda 1   Scheda 2   Scheda 3   Scheda 4

☒ Abilita scheda di rete

Connessa a: Scheda con bridge

Nome: Intel(R) Wi-Fi 6 AX200 160MHz

▶ Avanzate

Kali-brridged [In esecuzione] - Oracle VM VirtualBox

File   Macchina   Visualizza   Inserimento   Dispositivi   Aiuto

kali@kali: ~/Desktop

```
(kali@kali)-[~/Desktop]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.1.119 netmask 255.255.252.0 broadcast 10.0.3.255
    inet6 fe80::a00:27ff:fe21:1eb3 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:21:1e:b3 txqueuelen 1000 (Ethernet)
    RX packets 122971 bytes 18845603 (17.9 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 75719 bytes 5807814 (5.5 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
Message: Method Not Allowed.
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 5716 bytes 423388 (413.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5716 bytes 423388 (413.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali@kali)-[~/Desktop]
$
```

# Mythic Server Setup

Ci colleghiamo al nuovo server tramite PowerShell in SSH, aggiorniamo i repository e installiamo i prerequisiti per **Mythic**. Successivamente procediamo con l'installazione di Docker Compose e Make. Scarichiamo il repository di Mythic da GitHub, entriamo nella cartella appena clonata e utilizziamo il comando make, che esegue le operazioni necessarie per configurare ed eseguire Mythic, dopo che Docker e Docker Compose sono stati installati dallo script.

```
PS C:\Windows\system32> ssh root@155.152.152.152
The authenticity of host '155.152.152.152' is not yet known.
ECDSA key fingerprint is SHA256:155.152.152.152
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '155.152.152.152' (ECDSA) to the list of known hosts.
root@155.152.152.152:~#
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-122-generic x86_64)
```

```
root@Mythic:~# apt-get update && apt-get upgrade -y
```

```
root@Mythic:~# apt install docker-compose
```

```
root@Mythic:~# git clone https://github.com/its-a-feature/Mythic
Cloning into 'Mythic'...
remote: Enumerating objects: 25621, done.
remote: Counting objects: 100% (1226/1226), done.
remote: Compressing objects: 100% (374/374), done.
Receiving objects: 51% (13067/25621), 274.81 MiB | 30.89 MiB/s
```

```
root@Mythic:~# cd Mythic/
root@Mythic:~/Mythic# ls
CHANGELOG.MD      install_docker_kali.sh  Makefile          nginx-docker
documentation-docker  install_docker_ubuntu.sh  Mythic_CLI        postgres-docker
grafana-docker      InstalledServices       mythic-docker      postgres-exporter-docker
hasura-docker        jupyter-docker          mythic-react-docker  prometheus-docker
install_docker_debian.sh  LICENSE                 MythicReactUI      rabbitmq-docker
root@Mythic:~/Mythic# ./install_docker_ubuntu.sh
```

# Mythic Server Setup

```
root@Mythic:~/Mythic# make
cd Mythic_CLI && make build_linux && mv mythic-cli ../
make[1]: Entering directory '/root/Mythic/Mythic_CLI'
docker run -v /root/Mythic/Mythic_CLI/copy_file/:/copy_file/ --rm ghcr.io/its-a-feature/mythic_cli:v3.3.0.21 sh -c "cp /
mythic-cli_linux /copy_file/mythic-cli"
docker: Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?.
See 'docker run --help'.
make[1]: *** [Makefile:14: copy_binary_linux] Error 125
make[1]: Leaving directory '/root/Mythic/Mythic_CLI'
make: *** [Makefile:5: linux] Error 2
root@Mythic:~/Mythic#
```

Poi avviamo la **CLI** di **Mythic** per poter accedere al servizio e gestire le operazioni.

```
root@Mythic:~/Mythic# ./mythic-cli start
2024/10/04 15:50:12 [-] Error while reading in docker-compose file: Confi
thic]"
2024/10/04 15:50:12 [+] Successfully created new docker-compose.yml file.
2024/10/04 15:50:12 [+] Added mythic_postgres to docker-compose
2024/10/04 15:50:12 [+] Added mythic_react to docker-compose
2024/10/04 15:50:12 [+] Added mythic_server to docker-compose
2024/10/04 15:50:12 [+] Added mythic nginx to docker-compose
```

# Mythic Server Setup

Prima di accedere al servizio modificheremo il firewall di **Mythic** in modo da poter accedere solo dal mio PC. Andrò in 'Firewall'-'>'Manage'-'>'Add Firewall Group' con la seguente regola:

accept

TCP

1 - 65535

[redacted]/32



Farò lo stesso con gli IP del **Windows Server** e dell'**Ubuntu Server** ed applichiamo le regole al server **Mythic**

[Overview](#) [Usage Graphs](#) [Settings](#) [Snapshots](#) [Backups](#) [User-Data](#) [Tags](#) [DDOS](#)

IPv4

IPv6

VPC 2.0

Firewall

Custom ISO

Firewall (Manage)

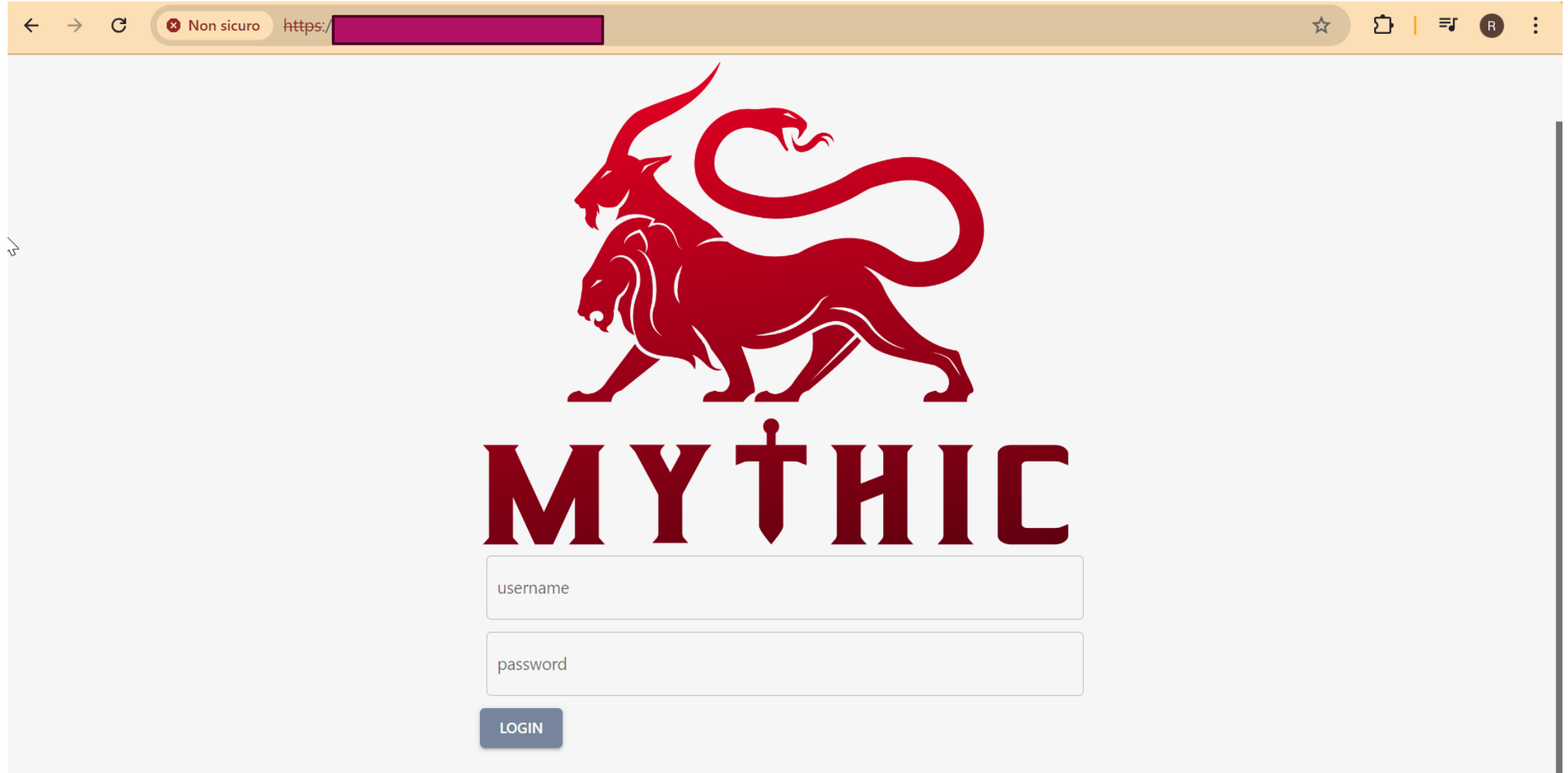
Firewall

1ec5a1a3-26bd-4dc2-b39d-41d775ec7b36: Mythic-Firewall

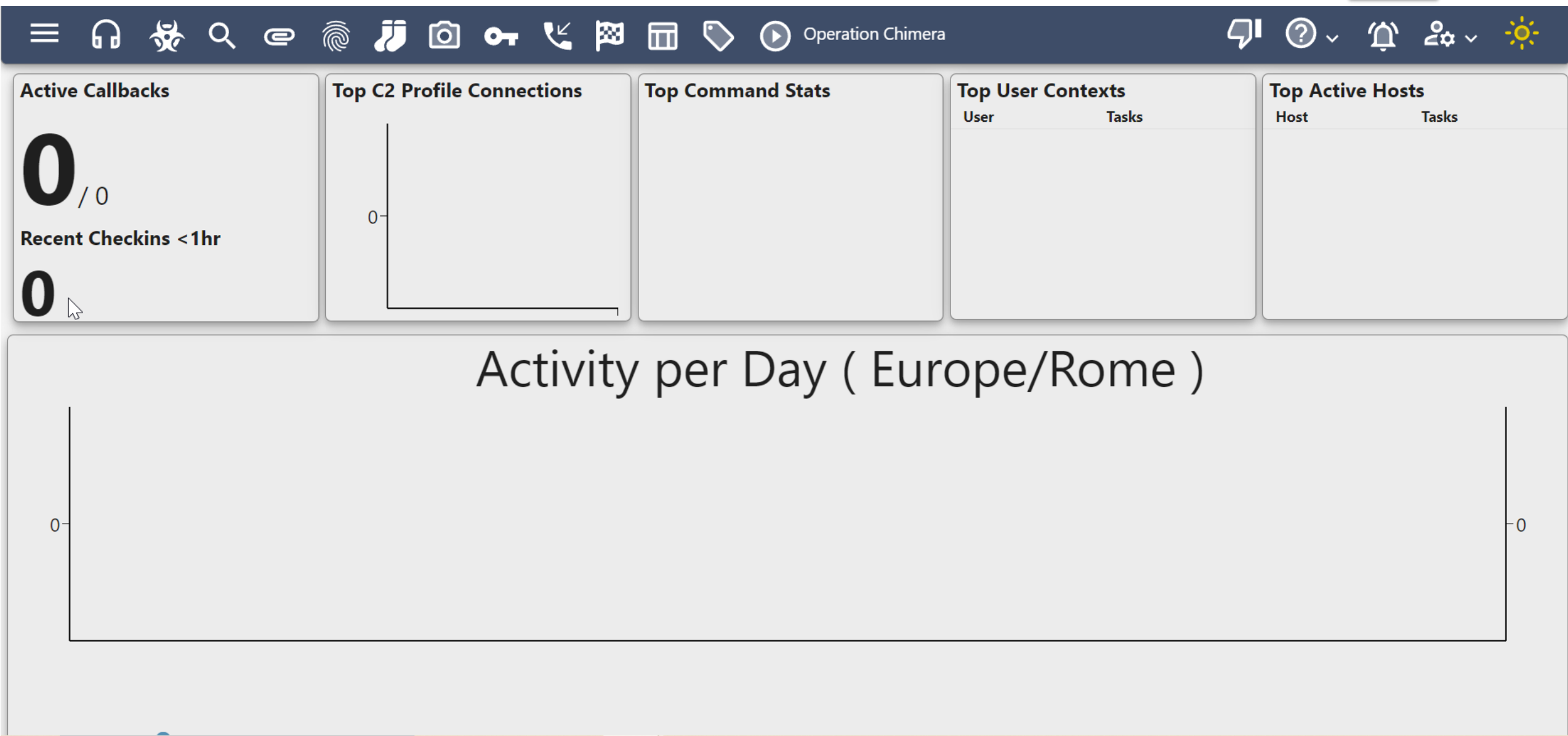
Update Firewall Group

Proviamo a collegarci alla Web UI di **Mythic** con l'username *mythic\_admin* e la password, che possiamo trovare nel file `.env` nella cartella di **Mythic**

```
JUPYTER_USE_VOLUME="false"  
JWT_SECRET="3cx2Q6heamZ1 [REDACTED]"  
MYTHIC_ADMIN_PASSWORD="2 [REDACTED]"  
MYTHIC_ADMIN_USER="mythic_admin"
```



L'interfaccia si presenta nel seguente modo:





# Mythic Server Setup

## Funzionalità principali della Web UI

### 1.Callbacks

Mostra lo stato attuale delle callback attive e le azioni intraprese. In questo esempio, nessuna callback attiva.

In **Mythic**, una **callback** si riferisce alla comunicazione stabilita tra un sistema compromesso (un host target) e il **C2 server**. Una volta che un **payload** viene eseguito su un sistema vittima, esso invia una richiesta al server Mythic, avviando una **callback**, che permette al server di impartire comandi e ricevere risposte dall'host compromesso.

Per ogni callback attiva, puoi anche vedere lo stato corrente della connessione: per esempio, se è attiva, se è in attesa di nuovi comandi, o se ci sono errori. È riportata anche l'ultima interazione, con data e ora della comunicazione più recente tra il server C2 e la macchina infetta. Un'altra informazione utile riguarda gli operatori che hanno eseguito azioni su quella callback, mostrando chi ha impartito comandi o monitorato la situazione.

# Mythic Server Setup

## 2.Icona Cuffie

Accesso ai servizi di **Command and Control (C2)**. Permette di configurare e gestire i servizi che coordinano le comunicazioni con gli agenti compromessi.

Ad esempio, potremmo avere un servizio HTTP configurato per far sembrare la comunicazione tra l'agente e il server un normale traffico web. Accedendo alla sezione C2 tramite l'icona **Cuffie**, puoi modificare le impostazioni di questo servizio, cambiare porte, monitorare i pacchetti o persino gestire più protocolli contemporaneamente.

## 3.Payloads

Creazione e gestione dei payload utilizzabili in attacchi. La UI fornisce esempi di azioni possibili tramite i payload, come l'esecuzione di comandi o la raccolta di informazioni.

# Mythic Server Setup

## Tabs principali

### 4.Search Tab

Qui possiamo cercare tra i **callback**, i **task** assegnati e altre attività.

**Keylogger**: Visualizzazione dei dati registrati dal keylogger, utile per ottenere credenziali o informazioni sensibili.

### 5.Files

Archivio per file caricati e ricevuti dagli agenti, con funzioni per l'upload e il download di dati tra la macchina compromessa e il server C2.

### 6.Artifacts

Visualizzazione e gestione degli artefatti raccolti durante le operazioni. Questi includono informazioni utili per l'analisi post-attacco o per la fase di investigazione.

### 7.Proxies for the SOC

Strumenti per il **SOC** (Security Operations Center), utili per monitorare e investigare sugli eventi di sicurezza legati a Mythic.

# Mythic Server Setup

## Altre Funzionalità

### 8.Report

Generazione di report dettagliati su tutte le operazioni eseguite, utile per il team di sicurezza per documentare l'attività e trarre conclusioni.

### 9.MITRE ATT&CK Mapping

Mythic permette il **mapping delle tattiche e tecniche ATT&CK** utilizzate durante le operazioni, fornendo una visione strutturata degli attacchi e delle contromisure. Questo è utile per:

**Classificare le azioni** rispetto al framework ATT&CK.

Pianificare nuove operazioni basate sulle tecniche più efficaci.