

# Terraform pour les opérations OffSec

Déploiement automatisé d'infrastructures de phishing et de C2 avec Terraform

Par: Martin Dubé

HackerspaceQuébecSec - Jeudi 26 Septembre 2024





# Ordre du jour

- 1 | INTRO
- 2 | TERRAFORM
  - A. Paradigme
  - B. Syntaxe
- 3 | OPÉRATIONS OFFSEC
  - A. Phishing
  - B. C2
- 4 | DEMO 

---

- 5 | CONCLUSION / QUESTIONS

# Intro



## LE DÉBUT D'UNE GRANDE AVENTURE

Cybersécurité Corsek Inc.

Déjà 9 mois d'existence!



## BACKGROUND

- 15 ans d'expérience en TI/Sécurité, 10 ans en Sécurité Offensive
- Impliqué dans le Hackfest de 2010 à 2017
- OSCP, OSCE, GREM, GCIH, GSEC



## INTÉRÊTS

- Course à pied - Premier marathon dans 10 jours!
- *Growth Mindset*
- *Woodworking, BBQ*



**CORSEK**  
CYBERSECURITY

# Origine de cette présentation





# Programmation déclarative

*La programmation déclarative est un paradigme de programmation qui consiste à créer des applications sur la base de composants logiciels.*

---

En d'autres mots, on décrit le *quoi*, et non le *comment*.

Exemple: HTML est déclaratif tandis que Python est impératif (ou procédurale).



# Déterminisme

*Le **déterminisme** est une théorie philosophique selon laquelle chaque événement, en vertu du **principe de causalité**, est déterminé par les événements passés conformément aux lois de la nature.*

---

Bref, un système pour lequel les mêmes entrées produisent toujours exactement les mêmes sorties

## Qu'est-ce que Terraform?

- ***INFRASTRUCTURE AS CODE***

Codifier l'infrastructure, facilitant le contrôle des versions, la collaboration et le déploiement automatisé.

- ***COMPATIBILITÉ MULTI-CLOUD***

Support d'un large éventail de fournisseurs de services cloud tels qu'AWS, Azure, Google Cloud et Proxmox.

- ***SYNTAXE DÉCLARATIVE***

Permet de définir l'état souhaité de l'infrastructure, sans s'occuper des étapes nécessaires pour la créer, la mettre à jour ou la détruire .

- ***CONCEPTION MODULAIRE***

Permet de décomposer l'infrastructure en composants réutilisables, améliorant ainsi la maintenabilité et l'évolutivité.

- ***GESTION DES ÉTATS***

Surveille fidèlement l'état actuel de l'infrastructure, lui permettant de planifier et d'appliquer efficacement les changements, tout en détectant et en résolvant les écarts.

# Impératif vs Déclaratif

	Impératif	Déclaratif
Exemple de language	Bash, C, Java, Python,	HTML, SQL, Terraform
Infrastructure: Créer	Un script	Une configuration
Infrastructure: Mettre à jour	Un autre script	La même configuration
Infrastructure: Inventorier	Un autre script	La même configuration
Infrastructure: Détruire	Un autre script	La même configuration
<i>Troubleshooting</i>	Relativement simple	Plus difficile

# Structure de fichiers

- **\*.TF**
  - Les noms de fichiers n'ont pas d'importance. Convention seulement.
  - Contient l'essentiel: **Ressource, Data, Variable, Output**

- **MODULES**

Regroupement de fichiers **\*.tf**

✓ WORK [DEV CONTAINER]

- > .terraform
- ✓ modules
  - > mythic-c2-server
  - ✓ phishing-server
    - > scripts
    - ✗ dns.tf
    - ✗ firewall.tf
    - ✗ main.tf
    - ✗ output.tf
    - ✗ secrets.tf
    - ✗ variables.tf
  - > sendgrid-authentication
  - > postfix-docker

✗ .terraform.lock.hcl

✗ main.tf

```
15 √ locals {  
16   |   project_name = "some-project-id"  
17 }  
18  
19 √ variable "gce_zone" {  
20   |   type      = string  
21   |   default   = "northamerica-northeast1-a"  
22 }  
23  
24 √ data "google_secret_manager_secret_version" "sendgrid"  
25   |   secret = "sendgrid-apikey"  
26 }  
27  
28 √ resource "google_compute_instance" "vm" {  
29   |   name      = "myvm"  
30   |   project   = local.project_name  
31   |   zone      = var.gce_zone  
32  
33 √   network_interface {  
34     |   network = google_compute_network.network.name  
35  
36 √     access_config {  
37       |   nat_ip = google_compute_address.nat_ip.address  
38     }  
39   }  
40  
41   #metadata_startup_script = file("scripts/startup.sh")  
42   metadata_startup_script = templatefile("${path.module}/scripts/startup.sh", {  
43     |   ...  
44   })
```

## Terraform

# Configuration

- **LOCALS**

Variables locales. Temporaire.

- **VARIABLE**

Variables qui peuvent être paramétré par variable d'environnement ou CLI.

- **DATA**

Ressource **déjà créé** dont on veut utiliser dans notre configuration.

- **RESOURCE**

Ressource à créer tel qu'une **VM, IP publique, règle de pare-feu, entrée DNS**.

# Terraform

# Utilisation

- **TERRAFORM VALIDATE**

Vérifier que la syntaxe est ok.

- **TERRAFORM PLAN**

Générer un plan d'exécution qui permet de **visualiser** les changements à effectuer.

- **TERRAFORM APPLY**

Appliquer un plan d'exécution

---

- **TERRAFORM Taint**

Assumer qu'une ressource est dégradé ou **endommagé**.

- **TERRAFORM DESTROY**

Détruire les resources gérés par la configuration courante.

```
Terraform used the selected providers to generate the following plan. Resource actions are indicated with the following symbols:
```

```
+ create
```

```
Terraform will perform the following actions:
```

```
# module.c2.google_compute_address.nat_ip will be created
+ resource "google_compute_address" "nat_ip" {
    + address          = (known after apply)
    + address_type     = "EXTERNAL"
    + creation_timestamp = (known after apply)
    + effective_labels = {
        + "goog-terraform-provisioned" = "true"
    }
    + id               = (known after apply)
    + label_fingerprint = (known after apply)
    + name             = "c2-nat-ip"
    + network_tier     = (known after apply)
    + prefix_length    = (known after apply)
    + project          = "quebecsec-2024"
    + purpose          = (known after apply)
    + region           = (known after apply)
    + self_link         = (known after apply)
    + subnetwork        = (known after apply)
    + terraform_labels = {
        + "goog-terraform-provisioned" = "true"
    }
    + users            = (known after apply)
}
```

```
1  
2  
3 data "google_secret_manager_secret_version" "postfix_fullchain_data"  
4   secret = "postfix-fullchain"  
5 }  
  
6 data "google_secret_manager_secret_version" "postfix_privkey_data" {  
7   secret = "postfix-privkey"  
8 }  
  
9 data "google_secret_manager_secret_version" "email_password_data" {  
10  secret = "emails-password"  
11 }  
  
12 data "google_secret_manager_secret_version" "sendgrid_apikey_data" {  
13  secret = "sendgrid-apikey"  
14 }  
  
15 resource "google_compute_instance" "vm" {  
16   name      = "myvm"  
17   project   = "some-project-id"  
18   zone      = "northamerica-northeast1-a"  
  
19   #metadata_startup_script = file("scripts/startup.sh")  
20   metadata_startup_script = templatefile("${path.module}/scripts/sta  
21   {  
22     fullchain      = data.google_secret_manager_secret_version.pc  
23     privkey        = data.google_secret_manager_secret_version.pc  
24     email_password = data.google_secret_manager_secret_version.em  
25     sendgrid_apikey = data.google_secret_manager_secret_version.se  
26   }  
  
27   # gcloud compute machine-types list | grep micro | grep us-central  
28   # e2-micro / 2 / 1.00  
29   # f1-micro / 1 / 0.60  
30   # gcloud compute machine-types list | grep small | grep us-central  
31   # e2-small / 2 / 2.00
```

## Terraform

# Meilleures pratiques

## Gestion des secrets

### ✓ VOÛTE DE SECRET

Utiliser les secrets en transit seulement.

### ✓ SSO

Favoriser SSO avant un jeton d'API.

### ✓ VARIABLES D'ENVIRONNEMENTS

Utile pour CI/CD, mais moins convivial pour développer.

### ✗ DANS LE CODE

Ne passez pas GO, ne réclamez pas 200\$.

.devcontainer > 🛠 Dockerfile

```
1  FROM mcr.microsoft.com/devcontainers/base:alpine-3.20
2
3  ARG PRODUCT=terraform
4  ARG VERSION=1.9.6
5
6  RUN apk add --update --virtual .deps --no-cache gnupg &&
7      cd /tmp && \
8      wget https://releases.hashicorp.com/${PRODUCT}/${VERSION}/
9      ${PRODUCT}_${VERSION}_linux_amd64.zip && \
10     wget https://releases.hashicorp.com/${PRODUCT}/${VERSION}/
11     ${PRODUCT}_${VERSION}_SHA256SUMS && \
12     wget https://releases.hashicorp.com/${PRODUCT}/${VERSION}/
13     ${PRODUCT}_${VERSION}_SHA256SUMS.sig && \
14     wget -qO- https://www.hashicorp.com/.well-known/pgp-ke
15     gpg --import && \
16     gpg --verify ${PRODUCT}_${VERSION}_SHA256SUMS.sig ${PI
17     {VERSION}_SHA256SUMS && \
18     grep ${PRODUCT}_${VERSION}_linux_amd64.zip ${PRODUCT}_
19     ${VERSION}_SHA256SUMS | sha256sum -c && \
unzip /tmp/${PRODUCT}_${VERSION}_linux_amd64.zip -d /1
mv /tmp/${PRODUCT} /usr/local/bin/${PRODUCT} && \
rm -f /tmp/${PRODUCT}_${VERSION}_linux_amd64.zip ${PRO
${VERSION}_SHA256SUMS ${VERSION}/${PRODUCT}_${VERSION}
_SHA256SUMS.sig && \
apk del .deps
ENTRYPOINT ["/bin/bash"]
```

## Terraform

# Meilleures pratiques

## Déterminisme

### ✓ TERRAFORM

- Les MAJ de terraform ne sont pas toujours rétrocompatibles.

### ✓ DOCKERFILE ET IMAGES DOCKER

- Réutiliser des images docker est plus déterministe que de bâtir à chaque fois.
- Éviter d'utiliser le tag **latest** (sauf si risque accepté).
- Préciser la version. Ex: **terraform:1.9.6**.
- Attention à: **apt, pkg, pip** (sauf si risque accepté).

### ✓ PYTHON

- Préciser la version:  
**pip install prettytable=3.11.0**

### ✗ COMPROMIS: SÉCURITÉ

- MAJ régulièrement et manuellement de façon contrôlé. Dependabot, Renovate, Snyk, Tests unitaires, Processus de *Gating*.

# Besoins en infrastructure



## ÉVOLUTIVITÉ ET FLEXIBILITÉ

Chaque opération est unique.

- Durée
- Acteur simulé
- Nombre de scénarios
- Domaines

## OPSEC

La configuration doit suivre les standards de l'équipe.

- Accès des opérateurs
- Filtrage SMTP
- Outils prêt à utilisation
- Redirecteurs

## EFFICIENCE

Maximiser l'opération

- Concentrer l'effort sur ce qui a de la valeur: TTPs et scénario
- Contrôler l'expositions de l'infrastructure
- Réduire les coûts

# Phishing

Configuration souhaitée

## VM

- IP publique (NAT)
- Un réseau dédié
- Clé SSH par défaut

## FIREWALL

- Ports 993, 587, etc.

## MISC

- Comptes par défaut

## POSTFIX (DOCKER)

- Envoyer des courriels
- Filtrer des entêtes
  - /^Received::.\*with ESMTP/
  - /^X-Originating-IP:/
  - /^X-Mailer:/
  - /^User-Agent:/

## DOVECOT (DOCKER)

- Recevoir des courriels

## DNS

- Sélectionner/Acheter un domaine
- TXT: SPF, DMARC
- A
- CNAME

## BONUS

- Sendgrid Authentication
- Mass-phish avec Gophish

## C2

### Configuration souhaitée

#### VM

- IP publique (NAT)
- Un réseau dédié
- Clé SSH par défaut

#### FIREWALL

- Ports 22, 80, 443

#### MISC

- Comptes par défaut

#### MYTHIC C2 (DOCKER)

- Installer et Configurer
  - Installer le profile HTTP
  - Installer l'agent Apollo
  - Installer l'agent Thanatos
- Sécuriser
  - Allowed IP block

#### CERTIFICATS TLS

- *cron.monthly*

#### DNS

- Sélectionner/Acheter un domaine
- A (ex. c2.quebecsec.xyz)

#### BONUS

- Redirecteur
- Custom Agents

# Demo



- vscode
- devcontainer
- Préparation du
- projet

- Examiner le
- code

- Déployer
- Tester

- Déployer
- Tester

