WAAP Lab guide - Terraform Version 1.0

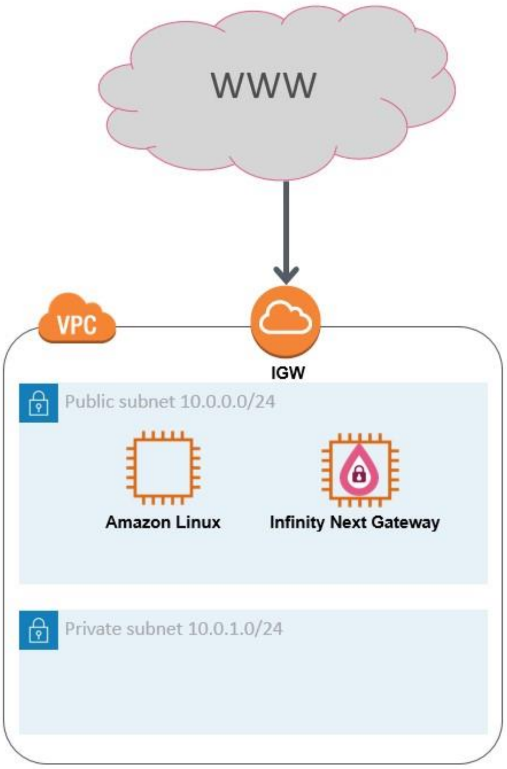
Amit Schnitzer DevSecOps 30/10/2020
CB Currier SE 12/1/202 – Terraform Integration

Prerequisites

- AWS account
- Check Point Infinity Portal Account
- Terraform

For your convenience, the admin guide can be found here

Lab Topology



Prepare the lab environment

Login to the AWS console https://console.aws.amazon.com

1. If you do not have an “Access Key” then generate one:
    a. Open the IAM console at https://console.aws.amazon.com/iam/
    b. On the navigation menu, choose Users.
    c. Choose your IAM user name (not the check box).
    d. Open the Security credentials tab, and then choose Create access key.
    e. To see the new access key, choose Show
    f. To download the key pair, choose Download .
2. Accept “Infinity Next Gateway” Terms and conditions in the following link https://aws.amazon.com/marketplace/pp?sku=cvxnu9a4ric9yop0tp0wdistb

3. Update the file variables.tf
    a. aws_account_id = "000000000000"
    b. aws_access_key = "AAAAAAAAAAAAAAAAAAAA"
    c. aws_secret_key = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
4. Set the region you are working in:
    a. aws_region = "us-east-1"
5. Update variables.tf file with the path and name of your AWS EC2 public ssh key:
    a. See the following link if you need to create one : https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html
    b. public_key_path = "~/.aws/MYPUBKEY.pem"
    c. update the following with the name of the keypair from your AWS Account:
        i. ssh_key_name = "MYPUBKEY"
6. Generate WAAP agent token
    a. Login to Infinity Portal (portal.checkpoint.com)
    b. Create a Localhost Asset
        i. Go to ENVIRONMENT tab
        ii. Click on “New” to create a new asset
        iii. Choose a name for the new asset (i.e. Localhost)
        iv. Under “application URL” type http://localhost and click on “+”
        v. Click the “Reverse Proxy” tab
        vi. Enter an upstream URL (i.e. http://127.0.0.1)
        vii. Click “Save”
    c. Go to “ENFORCEMENT” tab
    d. Under “Profiles” tab Click on “new” to create a new agent
        i. Choose a name for the new agent (i.e. WAAPLAB)
        ii. Agent type – choose “Infinity Next Gateway”
        iii. Click on “Reverse Proxy” tab above
        iv. Make sure you check the previously create web asset (Localhost) Leave the rest with default values and click “Save”
    e. Click on “Enforce”
    f. Click on “Tokens” to generate a token (copy and save the token string as it will be used during gateway installation)
7. Update the variables.tf file:
    a. InfinityToken = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
    b. Update any other variables fields if necessary.
8. Init, plan and apply
    a. Run ‘terraform init’

b. Run ‘terraform plan’

c. Run ‘terraform apply’

   i. Enter ‘yes’

9. Once complete the Public IP addresses of the Server and the Infinity Next GW will be returned
   a. Browse to http://<Server public IP>:7070/install.php to test installation
      i. Click on “install”
      ii. Register a new user name and login

Configuring WAAP protection

1. Login to Infinity Portal (portal.checkpoint.com)
2. Under “My Services”, click “Infinity Next”
3. First lets, create an asset representing our web server
   a. Go to ENVIRONMENT tab and under “Assets” choose to create a new
   b. Enter a name for the asset (i.e bwapp)
   c. Under “Application URL” type is the following http://<gw public IP>
   Make sure to replace with the gateway’s public IP from the AWS console
   d. Click on “Reverse Proxy” tab on the left
   e. Under “Upstream URL” enter the following http://10.0.0.10:7070 If you’ve assigned a different IP to the web server, make sure to replace it accordingly
   f. Click on “Enforce"
4. Now let’s create a policy that will be pushed and enforced on our gateway
   a. Go to POLICY tab on the left
   b. Under Rules create on “New” to create a new rule
   c. Under “Assets & Zones” click on “+” choose “Assets” and make sure you check the Asset we’ve created on step 2 above
   d. Under “Practices” click on “+” and choose “Web application protection”
   e. Under “Triggers”, choose “Log”
   f. Once done, click on “ENFORCE” to push policy on our gateway

Testing our configuration

Let’s test our configuration

Use-case-1: SQL Injection Attack

1. *Launch SQL injection attack – NO CG-WAAP interference*
   Start by logging into our vulnerable website directly http://<server public IP address>:7070/ using the following credentials bee/bug
   a. On the top right corner choose “SQL Injection (GET/SEARCH)” and click on “Hack”
   b. Now, in the search field, type “iron” for example and click on search
   c. Now, leave the search field empty and click on “search”
   d. Next, run the following “iron' or 1=1 -- “ and click “Search”
2. *Launch SQL Injection attack – CG-WAAP enabled*
   a. Login to Infinity Portal (portal.checkpoint.com)
   b. Under “ENFORCEMENTS” section click “Profiles” on the left
   c. Click “WAAPLAB” agent to see its configuration on the right
   d. Under “REVERSE PROXY” remove “Localhost” tick and enable “bwapp” tick.
   e. On top middle click “ENFORCE” to push latest policy to our agent.

    f. Now, we’ll login the vulnerable website through our Infinity Next gateway http://<gateway public IP address>/login.php (same credentials as above) and repeat step 1a through 1d. As you can see, the actual attack is being blocked by our WAAP gateway

Use-case-2: Directory Traversal Attack

1. Launch Directory traversal attack (do this in both browser tabs and examine the differences) – *NO CG-WAAP interference*
    a. On the top right corner choose “Directory Traversal – Directories” and click on “hack” button
    b. Replace the word “documents” from the URL field with “/etc” and see what happens.
2. Test same steps with Infinity Gateway URL

Use-case-3: PHP Code Injection Attack

3. Launch PHP Code Injection (do this in both browser tabs and examine the differences)
    a. On the top right corner choose “PHP Code Injection”
    b. Click on “message” and change the URL by replacing the word “test” with “phpinfo()”
    c. Change phpinfo() with system(‘ls’)
    d. Change system(‘ls’) with exec('whoami');
4. Test same steps with Infinity Gateway URL
5. Inspect the logs on infinity portal

Advanced tips / commands

Number	What	Action	Output
1	Check that agent is installed and running	cpnano -s	Make sure agent is in running state, check last time it was successfully updated
2		docker ps	Check that “cp_nginx_gaia” container is running successfully
3	Check pushed configuration	/etc/cp/rpmanager/servers	Folder containing the actual compiled configuration
4	Check agent logs	docker logs <container name>	See logs generated by conainer
5	Change agent’s debug level	vi /dev/shm/cp_nano_http_attachment_conf change debug level from 2 to 0	

6	Install / uninstall / Debug agent	/opt/CPWAAP/agent/install-cp-nanoagent.sh --install --token <token here> / --uninstall / --debug-on	
7			

–