WAAP Lab guide

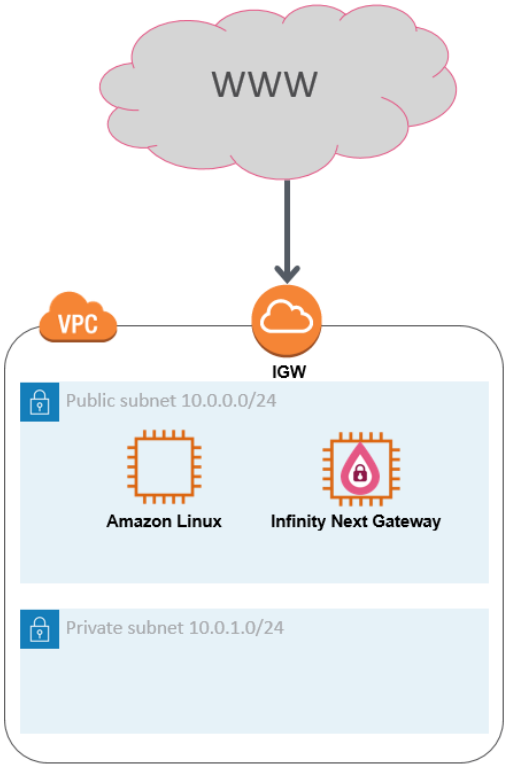
Amit Schnitzer
DevSecOps
30/10/2020

Prerequisites

- AWS account
- Check Point Infinity Portal Account

For your convenience, the admin guide can be found here

Lab Topology



Preparing your lab environment

Login to the AWS console (you can use your own or the one provided by instructor)

1. Use the “Launch VPC Wizard” and create a “VPC with a Single Public Subnet”
    a. IPv4 CIDR block - 10.0.0.0/16
    b. VPC Name “WAAP LAB” (or whatever you feel like calling it)
    c. Public subnet’s IPv4 CIDR - 10.0.0.0/24
    Leave the rest with default values
2. Add a private subnet
    a. Name Tag – Private Subnet

b. VPC – choose “WAAP LAB” VPC we’ve just created
c. IPv4 CIDR Block - 10.0.1.0/24

Leave the rest with default values

3. Create an EC2 key pair
    a. Key pair name – WAAP LAB
    b. File Format ppk
    c. Tags – you can leave empty
4. Launch Amazon Linux 2 AMI (HVM), SSD Volume Type
    a. EC2 Size – t2.micro is sufficient
    b. Network – WAAP LAB VPC
    c. Subnet – Public subnet
    d. Auto Assign Public IP – enable
    e. Network Interfaces --> Primary IP – 10.0.0.10
    f. Security Group
        - Add port 7070 from 0.0.0.0/0

    Leave the rest with default values and launch the instance
5. Installing Docker on our Linux EC2 server
    a. Run “sudo yum update –y”
    b. Run “sudo yum install -y docker”
    c. Run “sudo service docker start”
    d. Run “sudo usermod -a -G docker ec2-user”
    e. Run “Logout and log back in”
6. Launch our vulnerable web site
    a. Run “docker run -d -p 7070:80 raesene/bwapp”
    b. Run “docker ps” and make sure our container is running
    c. Browse to http://<instance public IP>:7070/install.php to test installation
    d. Click on “install”
    e. Register a new user name and login
7. Generate WAAP agent token
    a. Login to Infinity Portal (portal.checkpoint.com)
    b. Create a Localhost Asset
        - Go to ENVIRONMENT tab
        - Click on “New” to create a new asset
        - Choose a name for the new asset (i.e. Localhost)
        - Under “application URL” type http://localhost and click on “+”
        - Click the “Reverse Proxy” tab
        - Enter an upstream URL (i.e. http://127.0.0.1)
        - Click “Save”
    c. Go to “ENFORCEMENT” tab
    d. Under “Profiles” tab Click on “new” to create a new agent
        - Choose a name for the new agent (i.e. WAAPLAB)
        - Agent type – choose “Infinity Next Gateway”
        - Click on “Reverse Proxy” tab above
        - 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)
8. Install Infinity GW

a. Accept “Infinity Next Gateway” Terms and conditions in the following link https://aws.amazon.com/marketplace/pp?sku=cvxnu9a4ric9yop0tp0wdistb
b. Launch CloudGuard Infinity Next Gateway CFT into an existing VPC (#23 on the CFT list) from the following URL https://supportcenter.checkpoint.com/supportcenter/portal?eventSubmit_doGoviewsolutiondetails=&solutionid=sk111013
c. Choose “WAAP LAB” VPC
d. Choose the public and private subnets accordingly
e. Choose the key pair we previously created
f. Enter Password Hash (i.e. “1IFkdjsxm$4rreJ1DM4TFCqJ/F4I2xs/” for Cpwins1!)
g. Infinity Next Agent Token (paste the token generated on step 7.f. above)

Launch the installation and wait until CloudFormation finishes

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://<web 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

Finished already? Now launch the same on Azure.

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-nano-agent.sh --install --token <token here> / --uninstall / --debug-on	
7			

–

