Malware Protection Using an AWS Network Firewall

Lab overview

Malware, short for malicious software, refers to any intrusive software developed by cybercriminals (often called hackers) to steal data and damage or destroy computers and computer systems. Examples of common malware include viruses, worms, Trojan horses, spyware, adware, and ransomware.

Firewalls are like physical security walls situated between an organization's internal network and any external public networks such as the internet. The firewall protects an internal network from access by unauthorized users on an external network.

Users need access to the internet for business reasons, but they can inadvertently download malware, which can impact network and data security.

Malware threats can be present, and organizations can use various techniques and services to mitigate these threats (for example, firewalls, antivirus software, and user control best practice). This lab focuses on countermeasure techniques using a firewall.

Scenario

AnyCompany has hired you as a new security engineer, and the company has tasked you with hardening the company's security perimeter. There have been reports of users accidentally downloading malware after accessing specific websites. The IT team for AnyCompany has provided you with the URLs of the sites hosting the malware. It is your job to find a solution to mitigate access to these malicious actor files.

Objectives

After completing this lab, you should be able to:

- Update a network firewall
- Create a firewall rules group
- Verify and test that access to malicious sites is blocked

Duration

This lab requires approximately **45 minutes** to complete.

Lab environment

In this lab, you have a pre-configured **TestInstance** (Amazon Elastic Compute Cloud [Amazon EC2]) instance to use to test access to the website hosting malicious files. This is contained in a perimeter zone and separated from the rest of AnyCompany's important servers. You update the AnyCompany network firewall, create a rules group, and then attach that rules group to a firewall policy and the network firewall itself. You then log into the TestInstance and test the remediation.

All backend components, such as Amazon EC2, AWS Identity and Access Management (IAM) roles, and some AWS services, have been built into the lab already.

Task 1: Confirm Reachability

In this task, you log into the EC2 instance **TestInstance** that has been pre-configured during lab setup. From there, you issue a **wget** command to the malicious actor files that the IT team provided to you to confirm reachability.

wget is a free command-line utility and network file downloader.

6. From the Vocareum console page, choose the **AWS Details** button.
7. Next to **TestInstanceURL**, there is a link. Copy and paste the link into a new tab in your web browser.

   This link directly logs you into the TestInstance EC2 server via AWS Systems Manager Session Manager.
8. To change directories and view the current working directory, run the following commands:

   ```
   cd ~
   pwd
   ```

   The next step replicates how an end user would download a malicious file using a web browser. This action is simulated using the **wget** command on the malicious files in the command line.
9. In this protected lab environment, enter the following code and press Enter to download part of the malware:

   ```
   wget http://malware.wicar.org/data/js_crypto_miner.html
   ```
10. In this protected lab environment, enter the following code and press Enter to download the rest of the malware:

    ```
    wget http://malware.wicar.org/data/java_jre17_exec.html
    ```

Tip: Make sure to press Enter after copying and pasting each line of code to ensure that you run both lines of code.

These files are made specifically for anti-malware testing purposes and learning. Do not use them outside of this protected lab environment.

11. You should see an output similar to the following:

After connecting, you will get a confirmation with a "200 OK" status. This will download a js_crypto_miner.html file. This will also allow a successful download of a java_jre17_exec.html file.

12. To view the downloaded files, run the following command:

```
ls
```

The output should look like the following image:

```
sh-4.2$ ls
java_jre17_exec.html  js_crypto_miner.html
sh-4.2$
```

To confirm that the malware files were downloaded, the ls command was used. The output shows the java_jre17_exec.html and js_crypto_miner.html files that were downloaded.

Summary of task 1

In this task, you confirmed that the URL hosting the malware files is accessible through the current network and network firewall that AnyCompany is using. You used an isolated TestInstance EC2 instance to run commands and download the same malicious files that users downloaded. You now need to fix the AnyCompany network firewall to stop access to this site.

Task 2: Inspect the network firewall

In this task, you inspect the AWS Network Firewall **firewall** that was pre-configured during lab setup. Updating this firewall is the top priority that AnyCompany has issued to you as the new security engineer.

13. In the AWS Management Console, enter **VPC** in the search bar, and then choose **VPC**.
14. In the left navigation pane under **NETWORK FIREWALL**, choose **Firewalls**.
15. Choose **LabFirewall**, and read through the three steps in the **Overview** section.
16. In **Step 2: Configure the firewall policy**, choose the **LabFirewallPolicy** link to open the associated policy.

    A firewall policy defines the behavior of the firewall in a collection of stateless and stateful rule groups and other settings.
17. In the **Stateless default actions** section, choose **Edit**.
18. For **Stateless default actions**, configure the following options:
    - **Choose how to treat fragmented packets**: Choose **Use the same actions for all packets**.
    - **Action**: Choose **Forward to stateful rule groups**.
19. Choose **Save**.

    These settings now forward all packets to a stateful rule group for further inspection.

    A stateful rules engine inspects packets in the context of their traffic flow, gives you the ability to use more complex rules, and gives you the ability to log network traffic and AWS Network Firewall firewall alerts on traffic. Stateful rules consider traffic direction. The stateful rules engine might delay packet delivery to group packets for inspection.

    A stateless rules engine inspects each packet in isolation without regard to factors such as the direction of traffic or whether the packet is part of an existing, approved connection. This engine prioritizes the speed of evaluation.

Summary of task 2

In this task, you inspected the network firewall and updated the firewall policy. You then updated the firewall policy to forward all packets for stateful rule inspection.

Task 3: Create a firewall rule group

In this task, you create a network firewall rule group with rules that block access to the malicious URLs. You later attach this rule group to your firewall policy.

A network firewall rule group is a reusable set of criteria for inspecting and handling network traffic. You add one or more rule groups to a firewall policy as part of policy configuration. This rule group blocks access to the malicious actor URLs.

20. In the left navigation pane under **NETWORK FIREWALL**, choose **Network Firewall Rule Groups**.
21. Choose **Create Network Firewall rule group**.
22. In the **Create rule group** section, configure the following options:

- For **Rule group type**, choose **Stateful rule group**.
- For **Rule group format** choose **Suricata compatible rule string**.
- For **Rule evaluation order** choose **Action order**. Choose **Next**.
- In the **Rule group details** section, configure the following options:
    - **Name**: Enter `StatefulRuleGroup`
    - **Capacity**: Enter `100`
- Choose **Next**.

Intrusion prevention system (IPS) rules provide advanced firewall rules using Suricata rule syntax. Suricata is an open-source network IPS that includes a standard rule-based language for traffic inspection.

23. In the **Suricata compatible IPS rules** section, enter the following code into the text box:

```
drop http $HOME_NET any -> $EXTERNAL_NET 80 (msg:"MALWARE custom solution"; flow:
to_server,established; classtype:trojan-activity; sid:2002001;
content:"/data/js_crypto_miner.html";http_uri; rev:1;)

drop http $HOME_NET any -> $EXTERNAL_NET 80 (msg:"MALWARE custom solution"; flow:
to_server,established; classtype:trojan-activity; sid:2002002;
content:"/data/java_jre17_exec.html";http_uri; rev:1;)
```

The two Suricata rules that you added now block traffic that matches the **http_uri contents /data/js_crypto_miner.html** and **http_uri contents /data/js_crypto_miner.html** URLs when the traffic is initiated from the **LabVPC** to the public network.

24. Choose **Next** a few times to skip to the **Review and create** section. Finally **Create stateful rule group**.

Summary of task 3

In this task, you created a stateful network firewall rule group that uses Suricata rules. Once you attach this rule group to the network firewall, it blocks the malicious websites that AnyCompany users accessed.

Task 4: Attach a rule group to the network firewall

In this task, you attach the network firewall rule group that you created to the network firewall.

25. In the left navigation pane under **NETWORK FIREWALL**, choose **Firewalls**.
26. Choose **LabFirewall**.
27. Under **Associated firewall policy**, select the **LabFirewallPolicy**. Under **Stateful rule groups** select the dropdown list, and then choose **Add unmanaged stateful rule groups**.
28. Select the check box for **StatefulRuleGroup**, and then choose **Add stateful rule group**.

    At the top of the page, you should see a green **You successfully updated FirewallPolicy** banner.
29. Scroll to the **Stateful rule groups** section to see the successfully added firewall rule group.

Summary of task 4

You have attached the rule group to the firewall, which blocks attempts to access the malicious actor files hosted within the website.

Task 5: Validate the solution

In this task, you log back into the TestInstance to test that the network firewall properly blocks attempts to access the malicious website files.

30. In the AWS Management Console, enter **EC2** in the search bar, and then choose **EC2**.
31. In the left navigation pane, choose **Instances**.
32. Select the check box next for **TestInstance**, and then choose **Connect**.
33. Choose the **Session Manager** tab, and then choose **Connect**.
34. To change directories and view the current working directory, run the following commands:

    ```
    cd ~
    pwd
    ```

35. To try and access the first malicious file, run the following **wget** command.

    ```
    wget http://malware.wicar.org/data/js_crypto_miner.html
    ```

    The output should display the following:

    ```
    HTTP request sent, awaiting response...
    ```

    This output shows that the malware site and file are no longer accessible and have been successfully blocked by the network firewall.

36. Press Ctrl+c to stop the command.

37. To test access to the other malicious URL, run the following command:

```
wget http://malware.wicar.org/data/java_jre17_exec.html
```

    The output should display the following:

```
HTTP request sent, awaiting response...
```

38. Next, to remove the test malware files, run the following command:

```
rm java_jre17_exec.html js_crypto_miner.html
```

39. To confirm that the files were deleted, run the **ls** command:

```
ls
```

    You should see a blank output, which confirms that the files have been removed.

Summary of task 5

In this task, you verified that the network firewall has been updated and configured properly to block the malicious websites. You confirmed that access is blocked by logging into the TestInstance EC2 instance and running **wget** commands to these files. Users are now unable to access these malicious files from this website.