Team Task for Project

Monitoring Setup Instructions

- As the you attack a web server today, it will send all of the attack info to an ELK server.
- The following setup commands need to be run on the **Capstone** machine before the attack takes place in order to make sure the server is collecting logs.
- Be sure to complete these steps before starting the attack instructions.

Instructions

- Double click on the 'HyperV Manager' Icon on the Desktop to open the HyperV Manager.
- Choose the `Capstone` machine from the list of Virtual Machines and double-click it to get a terminal window.
- Login to the machine using the credentials: `vagrant:tnargav`
- Switch to the root user with `sudo su`

Setup Filebeat

Run the following commands:

- `filebeat modules enable apache`
- `filebeat setup`

The output should look like this:

Setup Metricbeat

Run the following commands:

- `metricbeat modules enable apache`
- `metricbeat setup`

The output should look like this:

Setup Packetbeat

Run the following command:

- `packetbeat setup`

The output should look like this:

Restart all 3 services. Run the following commands:

- `systemctl restart filebeat`
- `systemctl restart metricbeat`
- `systemctl restart packetbeat`

These restart commands should not give any output:

Once all three of these have been enabled, close the terminal window for this machine and proceed with your attack.

Attack!

Today, you will act as an offensive security Red Team to exploit a vulnerable Capstone VM.

You will need to use the following tools, in no particular order:

- Firefox
- Hydra
- Nmap
- John the Ripper
- Metasploit
- curl
- MSVenom

Setup

Your entire attack will take place using the `Kali Linux` Machine.

- Inside the HyperV Manager, double-click on the `Kali` machine to bring up the VM login window.
- Login with the credentials: `root:toor`

Instructions

Complete the following to find the flag:

- Discover the IP address of the Linux web server.
- Locate the hidden directory on the web server.
    - **Hint**: Use a browser to see which web pages will load, and/or use a tool like `dirb` to find URLs on the target site.
- Brute force the password for the hidden directory using the hydra command:
    - **Hint**: You may need to use `gunzip` to unzip `rockyou.txt.gz` before running Hydra.
    - **Hint**: `hydra -l <username> -P <wordlist> -s <port> -f -vV <victim.server.ip.address> http-get <path/to/secret/directory>`
- Break the hashed password with the Crack Station website or John the Ripper.
- Connect to the server via WebDav.
    - **Hint**: Look for WebDAV connection instructions in the file located in the secret directory. Note that these instructions may have an old IP Address in them, so you will need to use the IP address you have discovered.
- Upload a PHP reverse shell payload.
    - **Hint**: Try using your scripting skills! MSVenom may also be helpful.
- Execute payload that you uploaded to the site to open up a meterpreter session.
- Find and capture the flag.

After you have captured the flag, show it to your instructor.

Be sure to save important files (e.g., scan results) and take screenshots as you work through the assessment. You'll use them again when creating your presentation.

Day 2

Solution Guide: Incident Analysis with Kibana

Instructions: Investigating the Incident

Even though you already know what you did to exploit the target, analyzing the logs is still valuable. It will teach you:

- What your attack looks like from a defender's perspective.
- How stealthy or detectable your tactics are.
- Which kinds of alarms and alerts SOC and IR professionals can set to spot attacks like yours while they occur, rather than after.
- While going through the solution file, please note that the IP addresses here need to be replaced your machine's IP addresses.

Double-click the Google Chrome icon on the Windows host's desktop to launch Kibana. If it doesn't load as the default page, navigate to http://192.168.1.105:5601.

Start by creating a Kibana dashboard using the pre-built visualizations. Navigate to your home page, then scroll down to **Visualize and Explore Data** then **Dashboard**.

Click on **Create dashboard** in the upper left hand side. On the new page click on **Add an existing** to add the following existing reports:

- `HTTP status codes for the top queries [Packetbeat] ECS`
- `Top 10 HTTP requests [Packetbeat] ECS`
- `Network Traffic Between Hosts [Packetbeat Flows] ECS`
- `Top Hosts Creating Traffic [Packetbeat Flows] ECS`
- `Connections over time [Packetbeat Flows] ECS`
- `HTTP error codes [Packetbeat] ECS`
- `Errors vs successful transactions [Packetbeat] ECS`
- `HTTP Transactions [Packetbeat] ECS`

Your final dashboard should look similar to:

Next, get familiar with running search queries in the `Discover` screen with Packetbeat.

- On the Discover page, locate the search field.
- Start typing `source` and notice the suggestions that come up.
- Search for the `source.ip` of your attacking machine.
- Use `AND` and `NOT` to further filter you search and look for communications between your attacking machine and the victim machine.
- Other things to look for:
    - `url`
    - `status_code`
    - `error_code`

Some helpful searches include

- `http.response.status_code : 200`
- `url.path: /company_folders/secret_folder/`
- `source.port: 4444`
- `destination.port: 4444`

- `NOT source.port: 80 and NOT source.port: 443`

After you create your dashboard and become familiar with the search syntax, use these tools to answer the questions below:

1. Identify the Offensive Traffic

Identify the traffic between your machine and the web machine:

- Staring with a few searches in the 'Discover' area, we can find some interesting interactions.
- Run `source.ip: 192.168.1.90 and destination.ip: 192.168.1.105` in which the source IP is your Kali machine and your destination machine is your web server.
- Run `url.path: /company_folders/secret_folder/`.

When did the interaction occur?

- You know when the interaction happened so we will need to change the timeline that Kibana is searching to see that time period:

In your dashboard, look through the different panels and use this data to look through the results and notice the following interactions:

What responses did the victim send back?

- On our dashboard, we can see the top responses in the `HTTP status codes for the top queries [Packetbeat] ECS`

- We can see `401`, `301`, `207`, `404` and `200` as the top responses.
- We can also see with the `HTTP Error Codes [Packetbeat] ECS` panel:

What data is concerning from the Blue Team perspective?

- We can see a connection spike in the `Connections over time [Packetbeat Flows] ECS`

- We can also see a spike in errors in the `Errors vs successful transactions [Packetbet] ECS`

2. Find the Request for the Hidden Directory

In your attack, you found a secret folder. Let's look at that interaction between these two machines.

How many requests were made to this directory? At what time and from which IP address(es)?

- On the dashboard you built, a look at your `Top 10 HTTP requests [Packetbeat] ECS` panel:

- In this example we can see that this folder was requested `6,197` times.

Which files were requested? What information did they contain?

- We can see in the same panel that the file `connect_to_corp_server` was requested `3` times.

What kind of alarm would you set to detect this behavior in the future?

- We could set an alert that goes off for any machine that attempts to access this directory or file.

Identify at least one way to harden the vulnerable machine that would mitigate this attack.

- This directory and file should be removed from the server all together.

3. Identify the Brute Force Attack

After identifying the hidden directory, you used Hydra to brute-force the target server. Answer the following questions:

Can you identify packets specifically from Hydra?

- Yes, if you are using the search function `url.path: /company_folders/secret_folder/` will show you a few conversations involving this folder.
- In the `Discovery` page, search for: `url.path: /company_folders/secret_folder/`.

Look through the results and notice that `Hydra` is identified under the `user_agent.original` section:

How many requests were made in the brute-force attack? How many requests had the attacker made before discovering the correct password in this one?

- In the `Top 10 HTTP requests [Packetbeat] ECS` panel, we can see that the password protected `secret_folder` was *requested* `6209` times, but the file inside that directory was only requested `3` times. So, out of `6209` requests, only `3` were successful.

Note: Your results will differ.

Take a look at the `HTTP status codes for the top queries [Packetbeat] ECS` panel:

- You can see on this panel the breakdown of `401 Unauthorized` status codes as opposed to `200 OK` status codes.
- We can also see the spike in both traffic to the server and error codes.
- We can see a connection spike in the `Connections over time [Packetbeat Flows] ECS`

- We can also see a spike in errors in the `Errors vs successful transactions [Packetbet] ECS`

These are all results generated by the brute force attack with Hydra.

What kind of alarm would you set to detect this behavior in the future and at what threshold(s)?

- We could set an alert if `401 Unauthorized` is returned from any server over a certain threshold that would weed out forgotten passwords. Start with `10` in one hour and refine from there.
- We could also create an alert if the `user_agent.original` value includes `Hydra` in the name.

Identify at least one way to harden the vulnerable machine that would mitigate this attack.

- After the limit of 10 `401 Unauthorized` codes have been returned from a server, that server can automatically drop traffic from the offending IP address for a period of 1 hour. We could also display a lockout message and lock the page from login for a temporary period of time from that user.

4. Find the WebDav Connection

Use your dashboard to answer the following questions:

How many requests were made to this directory?

- We can again see in the `Top 10 HTTP requests [Packetbeat] ECS` panel that the webdav folder was directly connected and files inside were accessed.

- We can also see it in the pie charts:

Which file(s) were requested?

- We can see the passwd.dav file was requested as well as a file named `shell.php`

What kind of alarm would you set to detect such access in the future?

- We can create an alert anytime this directory is accessed by a machine *other* than the machine that should have access.

Identify at least one way to harden the vulnerable machine that would mitigate this attack.

- Connections to this shared folder should not be accessible from the web interface.
- Connections to this shared folder could be restricted by machine with a firewall rule.

5. Identify the Reverse Shell and meterpreter Traffic

To finish off the attack, you uploaded a PHP reverse shell and started a meterpreter shell session. Answer the following questions: Can you identify traffic from the meterpreter session?

- First, we can see the `shell.php` file in the `webdav` directory on the `Top 10 HTTP requests [Packetbeat] ECS` panel.

- Remember that your meterpreter session ran over port `4444`. Port `4444` is the *default* port used for meterpreter and the port used in all of their documentation. Because of this, many attackers forget to change this port when conducting an attack. You can construct a search query to find these packets.
- `source.ip: 192.168.1.105 and destination.port: 4444`

What kinds of alarms would you set to detect this behavior in the future?

- We can set an alert for any traffic moving over port `4444`.
- We can set an alert for any `.php` file that is uploaded to a server.

Identify at least one way to harden the vulnerable machine that would mitigate this attack.

- Removing the ability to upload files to this directory over the web interface would take care of this issue.

⚠ Important Checkpoint ⚠

At this time, you should have completed the following steps:

Step 1: Identify the Offensive Traffic.

Step 2: Find the Request for the Hidden Directory.

Step 3: Identify the Brute Force Attack.

Step 4: Find the WebDav Connection.

Step 5: Identify the Reverse Shell and meterpreter Traffic.

To complete the next part of the project, you should take screen shots that represent each of the issues listed in preparation of compiling them into a report.

Day 3 Activity File: Reporting

Congratulations! This week, you've worn two hats, playing the roles of attacker and defender. Don't underestimate the magnitude of this achievement: learning enough to infiltrate a machine and analyze data collected during an attack is a milestone that takes many professionals a long time to achieve.

Today, you'll take a break from flexing your technical skills and focus on communicating what you've learned during this project. In a real engagement, your client pays you not to break into their network, but to teach them how to protect it. This is why communication skills are vital in the cybersecurity field.

Therefore, you will summarize your work in a presentation containing the following sections:

- **Network Topology**: What are the addresses and relationships of the machines involved?
    - **Solution**: The following machines live on the network:
        - **Kali**: `192.168.1.90`
        - **ELK**: `192.168.1.100`
        - **Target**: `192.168.1.105`
- **Red Team**: What were the three most critical vulnerabilities you discovered? Choose the three vulnerabilities that *you* consider to be most critical.
    - **Solution**: While the web server suffers from several vulnerabilities, the three below are the most critical:
        - **Cryptographic Failures**: Exposure of the `secret_folder` directory and the `connect_to_corp_server` file compromised the credentials of the Web DAV folder. Cryptographic Failures is an OWASP Top 10 vulnerability.
        - **Unauthorized File Upload**: The web server allows users to upload arbitrary files — specifically, PHP scripts. This exposes the machine to the wide array of attacks enabled by malicious files.
        - **Remote Code Execution**: As a consequence of the unauthorized file upload vulnerability, attackers can upload web shells and achieve arbitrary remote code execution on the web server.
        - Additional severe vulnerabilities include:
            - Lack of mitigation against brute force attacks
            - No authentication for sensitive data, e.g., `secret_folder`
            - Plaintext protocols (HTTP and WebDAV)
- **Blue Team**: What evidence did you find in the logs of the attack? What data should you be monitoring to detect these attacks in the future?
    - **Solution**: A considerable amount of data is available in the logs. Specifically, evidence of the following was obtained upon inspection:
        - Traffic from attack VM to target, including unusually high volume of requests
        - Access to sensitive data in the `secret_folder` directory
        - Brute-force attack against the HTTP server
        - POST request corresponding to upload of `shell.php`

- **Unusual Request Volume**: Logs indicate an unusual number of requests and failed responses between the Kali VM and the target. Note that `401`, `301`, `207`, `404` and `200` are the top responses.

    - In addition, note the connection spike in the `Connections over time [Packetbeat Flows] ECS`, as well as the spike in errors in the `Errors vs successful transactions [Packetbet] ECS`

- **Access to Sensitive Data in `secret_folder`**: On the dashboard you built, a look at your `Top 10 HTTP requests [Packetbeat] ECS` panel. In this example, this folder was requested `6,197` times. The file `connect_to_corp_server` was requested `3` times.

- **HTTP Brute Force Attack**: Searching for `url.path: /company_folders/secret_folder/` shows conversations involving the sensitive data. Specifically, the results contain requests from the brute-forcing tool`Hydra`, identified under the `user_agent.original` section:

    - In addition, the logs contain evidence of a large number of requests for the sensitive data, of which only `3` were successful. This is a telltale signature of a brute-force attack. Specifically, the password protected `secret_folder` was requested `6209` times. However, the file inside that directory was only requested `3` times. So, out of `6209` requests, only `3` were successful.

- **WebDAV Connection & Upload of `shell.php`**: The logs also indicate that an unauthorized actor was able to access protected data in the `webdav` directory. The `passwd.dav` file was requested via `GET`, and `shell.php` uploaded via `POST`.

- **Mitigation**: What alarms should you set to detect this behavior next time? What controls should you put in place on the target to prevent the attack from happening?
    - **Solution**: Mitigation steps for each vulnerability above are provided below.
        - **High Volume of Traffic from Single Endpoint**
            - Rate-limiting traffic from a specific IP address would reduce the web server's susceptibility to DoS conditions, as well as provide a hook against which to trigger alerts against suspiciously suspiciously fast series of requests that may be indicative of scanning.
        - **Access to sensitive data in the `secret_folder` directory**
            - First, the `secret_folder` directory should be protected with stronger authentication. E.g., it could be moved to a server to which only key-based SSH access from whitelisted IPs is enabled.
            - Second, the data inside of `secret_folder` should be encrypted at rest.
            - Third, Filebeat should be configured to monitor access to the `secret_folder` directory and its contents.
            - Fourth, access to `secret_folder` should be whitelisted, and access from IPs not on this whitelist, logged.
        - **Brute-force attack against the HTTP server**
            - The `fail2ban` utility can be enabled to protect against brute force attacks.
        - **POST request corresponding to upload of `shell.php`**
            - File uploads should require authentication.
            - In addition, the server should implement an upload filter and forbid users from uploading files that may contain executable code.

Presentation Deliverables