

Misc Challenge #1

The hackermonkeys.com developer is known to make mistakes with his coding skills; can you find anything on the internet he may have accidentally leaked?

Solution

Overview

Contestants will perform Open-Source-Intelligence (OSINT) gathering of the 'hackermonkeys.com' developer. With references to a developer and his coding skills in the challenge itself, he will have his own GitHub account that contestants will have to find. A simple search query on <https://github.com> for 'hackermonkeys.com' will display a single user named Bubbles whose description indicates they work at 'hackermonkeys.com'. In addition, users may perform DNS reconnaissance against the 'hackermonkeys.com' domain and discover several references to the developer Bubbles, including his email. Should the players decide to email him, he will respond with an Out-of-Office email indicated he is on vacation but that he is looking for a new job and offers his Github.com profile link.

Within his two GitHub repositories, only one has the flag buried in his commit history. Bubbles accidentally pushed a code change with a base64 secret within the 'pewpew' repository that is now public. Contestants will have to dig through all 200+ commits manually or use one of many tools to help dig up secrets within code repositories. Accidental credential leakage within a development pipeline is a common vulnerability that can be exploited to gain access to sensitive systems or services.

After finding the secret, contestants will have to base64 decode the flag into:

monkeyCTF{8vfvwf3kaw4vxcz41d78vzxrltv73cw0}

TECHNICAL DETAILS

Browsing to GitHub.com and querying for 'hackermonkeys.com' will display a single user who matches the query.

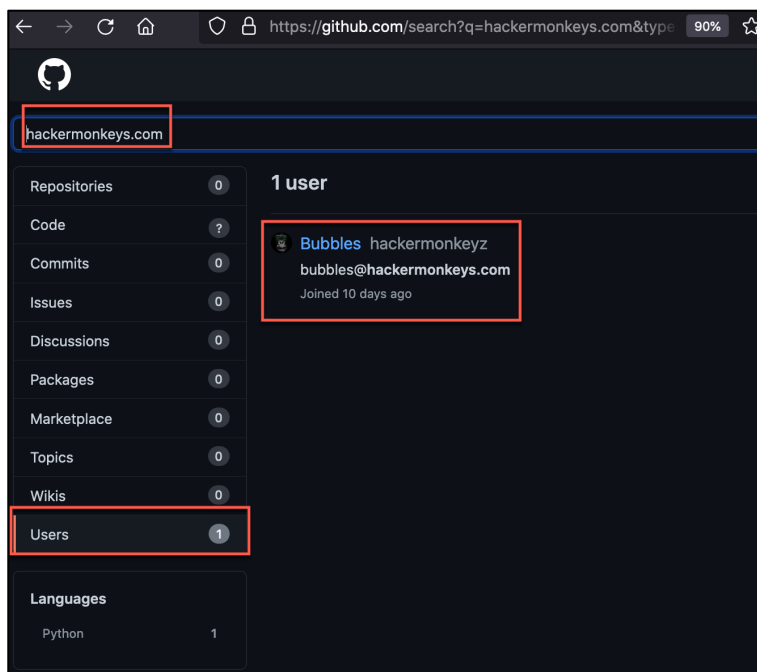


Figure 1: Searching GitHub

Contestants can perform reconnaissance to discover Bubbles email to send him a message.

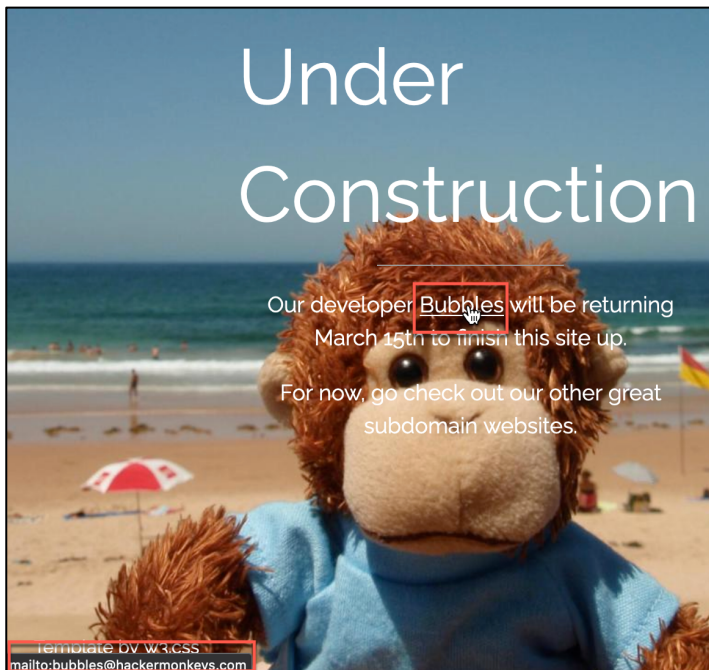


Figure 2: Bubbles email on hackermonkeys.com

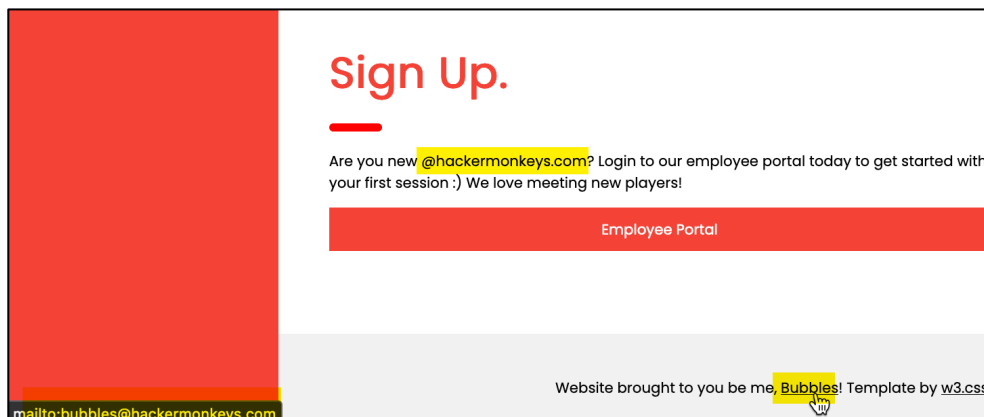


Figure 3: Bubbles email on dnd.hackermonkeys.com

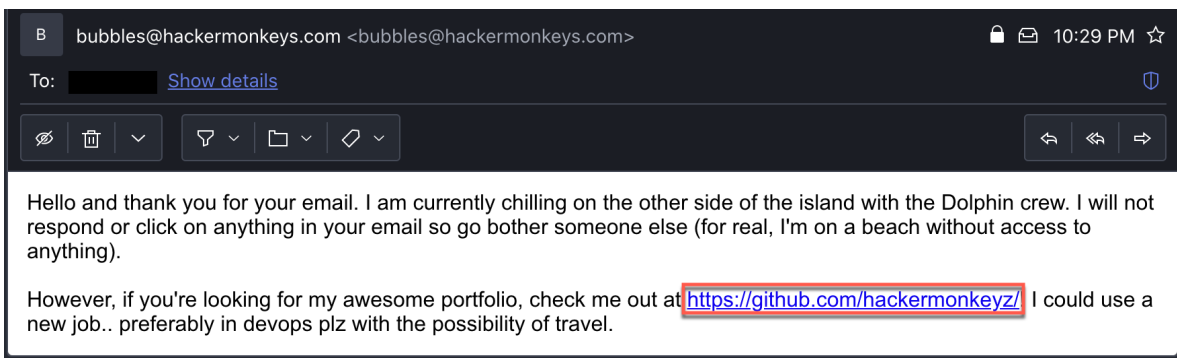


Figure 4: Bubbles GitHub link in autoreply email

Contestants will have to enumerate Bubbles' GitHub repositories. The 'scripts' repository has nothing of value, just old scripts Bubbles had saved in some old folders and with only 1 commit history. The 'pewpew' repository, however, has over 200 commits.

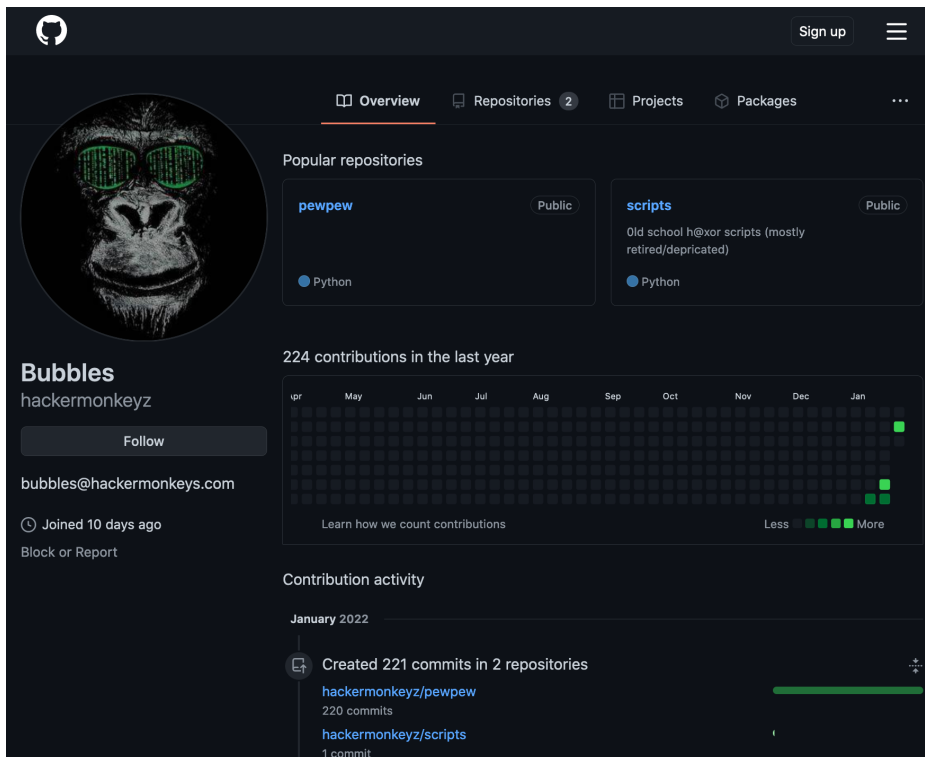


Figure 5: Bubbles GitHub repositories

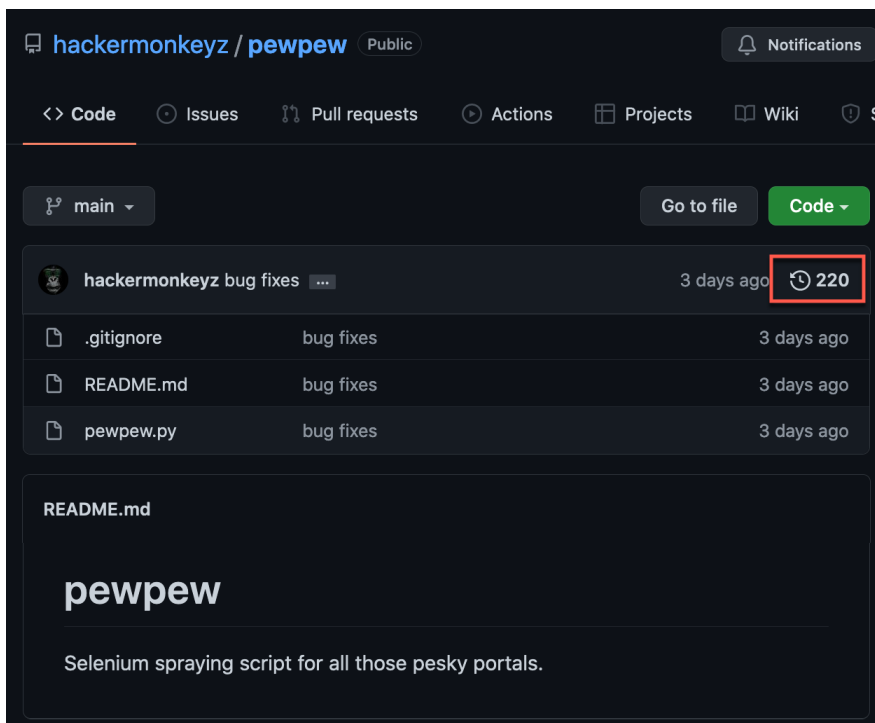
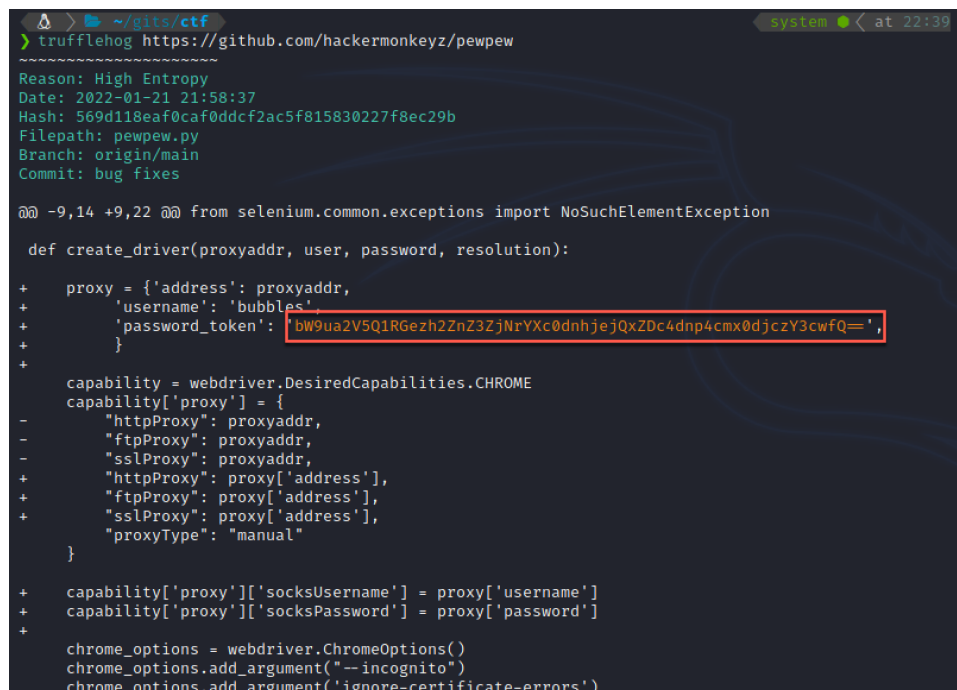


Figure 6: Large Git commit history

Running a tool like truffleHog¹ against the public repository (or could be cloned locally and regex'd through using other tools) reveals a 'password_token' parameter to a proxy Bubbles was using. The value is Base64 encoded. Decoding it, reveals the flag to be:

monkeyCTF{8vfvwf3kaw4vxcz41d78vzxrltv73cw0}



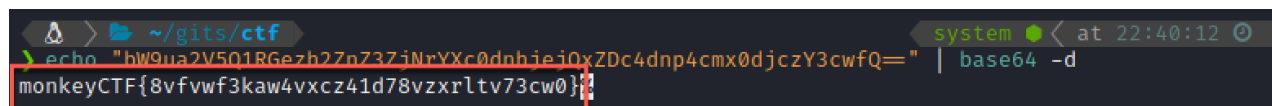
```
> trufflehog https://github.com/hackermoneyz/pewpew
Reason: High Entropy
Date: 2022-01-21 21:58:37
Hash: 569d118eaf0caf0ddcf2ac5f815830227f8ec29b
Filepath: pewpew.py
Branch: origin/main
Commit: bug fixes

@@ -9,14 +9,22 @@ from selenium.common.exceptions import NoSuchElementException

def create_driver(proxyaddr, user, password, resolution):
+   proxy = {'address': proxyaddr,
+           'username': 'bubbles',
+           'password_token': 'bW9ua2V5Q1RGezh2ZnZ3ZjNrYXc0dnhiejqxZDc4dnp4cmx0djcZy3cwFQ==',
+           }
+
+   capability = webdriver.DesiredCapabilities.CHROME
+   capability['proxy'] = {
-       "httpProxy": proxyaddr,
-       "ftpProxy": proxyaddr,
-       "sslProxy": proxyaddr,
+       "httpProxy": proxy['address'],
+       "ftpProxy": proxy['address'],
+       "sslProxy": proxy['address'],
+       "proxyType": "manual"
+   }

+   capability['proxy']['socksUsername'] = proxy['username']
+   capability['proxy']['socksPassword'] = proxy['password']
+
+   chrome_options = webdriver.ChromeOptions()
+   chrome_options.add_argument("--incognito")
+   chrome_options.add_argument('ignore-certificate-errors')
```

Figure 7: Scanning for secrets in the 'pewpew' repository



```
echo "bW9ua2V5Q1RGezh2ZnZ3ZjNrYXc0dnhiejqxZDc4dnp4cmx0djcZy3cwFQ==" | base64 -d
monkeyCTF{8vfvwf3kaw4vxcz41d78vzxrltv73cw0}
```

Figure 8: Base64 decoded flag

¹ <https://github.com/trufflesecurity/truffleHog>