

DateEscape: The Innovative New Way to Escape a Commitment

Jeffrey M. Dinackus

Northern Virginia Community College

DateEscape is the automated way to have a friend give you a phone call during a date so you can duck out with an excuse. It uses Python to take information and make Twilio API requests on behalf of the user to send text messages and calls directly to the user's phone. The call will play a recorded message from the user and the texts can be anything that is text. The result is a much more convenient way to automate the receiving of calls and texts. It is also highly customizable; API requests can be sent seconds or minutes apart to simulate a real interaction with some modification to the script. DateEscape can be used for any interaction if you know you may want to escape in advance or if can use your phone during the interaction. It relies on AWS EC2 for hosting the server that runs the Python script, after it is cloned from GitHub. This can be obtained for free by using an AWS free tier eligible t.2 micro instance and the script instructions are configured for Ubuntu. The user will also need an API key from Twilio and a Twilio phone number which can be obtained for free. The user is also recommended to save the Twilio phone number as a contact in their phone before the running of the script adding realism as the calls and texts will say the name of anything they enter! DateEscape can be found for free on my GitHub (Jeffrey Dinackus, 2023).

There are 3 main scripts used by DateEscape, `initial-setup.py`, `test-connection.py`, and `make-a-call.py`. Initial setup takes in the variables needed from the user, such as his or her phone number, his or her Twilio phone number, and API keys then it saves them to individual files. Then, it prints them so the user can check his or her inputs. `Test-connection.py` sends a text and a call to the user using his or her input. The user is free to edit his or her information anytime by changing the values in his or her files. If the user receives them, he or she is good to go. `Make-a-call.py` reads the inputs and creates a timer which counts down to when it is supposed to call based on user input. This is where the API requests are made. I did a lot of refactoring

throughout the project to make the project cleaner and more readable. For example, instead of having two API call sections, one for if the user enters “now” to be called now, and one for timed calls, combined into one with a function. This will make modifications to the script (the order or number of calls and texts) much less confusing for the user.

During development, the services I used were AWS EC2 (also required for running the program) and AWS CodeGuru Security (“Find Expensive Code – Amazon CodeGuru Security – AWS,” n.d.). The total time developing was at least 40 hours, as this project was originally a hackathon project and I worked almost nonstop on it for 2 days (Jeffrey Dinackus & Trombo, 2023). Post hackathon, the development was probably 10 more hours due to design, refactoring, documentation, and testing. Some major challenges include the referencing of the user information and API keys, which were originally desired to be environmental variables per the hackathon rules. Python had a hard time reading the environmental variables and later I decided that it was a better user experience for the user to be able to open a file and simply edit the information. The timer was a major hinderance in the original DateEscape, called QuickEscape. Due to the short time period of the hackathon, I was not able to complete it in time, it is fully working for the new project. The Python library time operates via seconds, but the input is minutes and hours, meaning a conversion was necessary. The original project also had to poll a MongoDB Atlas DB to receive the information needed to make the calls. The original script was continually polling the Atlas DB for new events that would take place within 10 seconds then activate. This was far from ideal. The reason for this was that there was a React front end which the user entered their information into. This front end has been completely scrapped in the current version for simplicity. The current setup for this project also isn’t ideal. It requires the creation of a Twilio account, verifying a phone number with Twilio, receiving a Twilio Phone

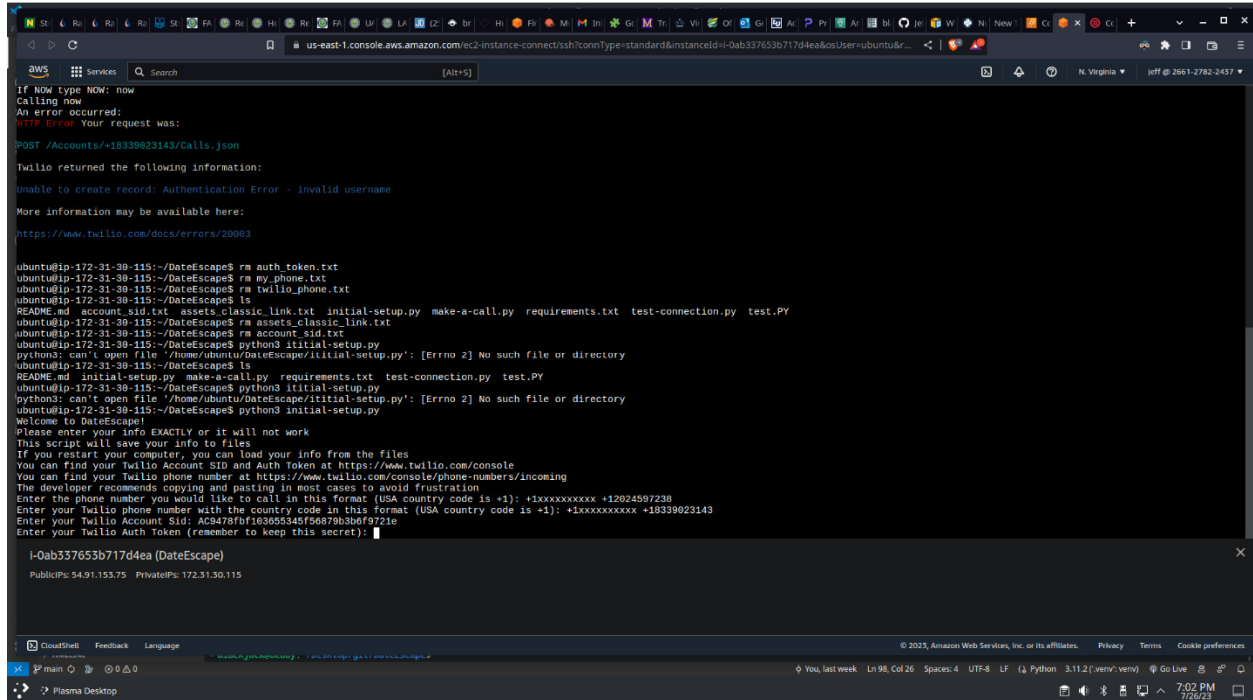
number, and probably some troubleshooting to get it all to work. I have done the heavy lifting of writing the API calls and structure. In the original deployment we would have had to charge the user for the API requests they make or require them to enter their API keys every time. In the current deployment, the user can run it for free or very cheap by getting his or her own API keys. I have written documentation to ease this. It is a one-time setup, after you get it working you can run it many times as long as you keep the instance running or save the script between uses. The user can create his or her desired script by copying and pasting the API requests he wishes to make. In the future I may want to create a front end and manage API keys for the user, like the original deployment.

AWS EC2 is required to make this project work and is much simpler to use than a traditional server. I also used AWS CodeGuru Security to do a security check of my code by uploading a zip of my project file a few times. I resolved one security flaw by adding some type checking of one of the variables. The other nine vulnerabilities only apply to Python 2, and as my project uses Python 3, are false positives. Screenshots of CodeGuru Security have been added to the Appendix of this paper.

References

- Jeffrey Dinackus. (2023, July 28). GitHub - JeffreyDinackus/DateEscape: This app allows you to make prerecorded calls and texts to yourself so you can get out of a date. Retrieved July 28, 2023, from <https://www.github.com/jeffreydinackus/dateescape>
- Jeffrey Dinackus, & Trombo, J. (2023, March 25). GitHub - JeffreyDinackus/QuickEscape.tech. Retrieved July 28, 2023, from <https://github.com/JeffreyDinackus/QuickEscape.tech>
- Find expensive code – Amazon CodeGuru Security – AWS. (n.d.). Retrieved from <https://aws.amazon.com/codeguru/>

Appendix



```
aws
Services
Search
[Alt+S]

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0ab337653b717d4ea&osUser=ubuntu&r...
N. Virginia jeff @ 261-2782-2437

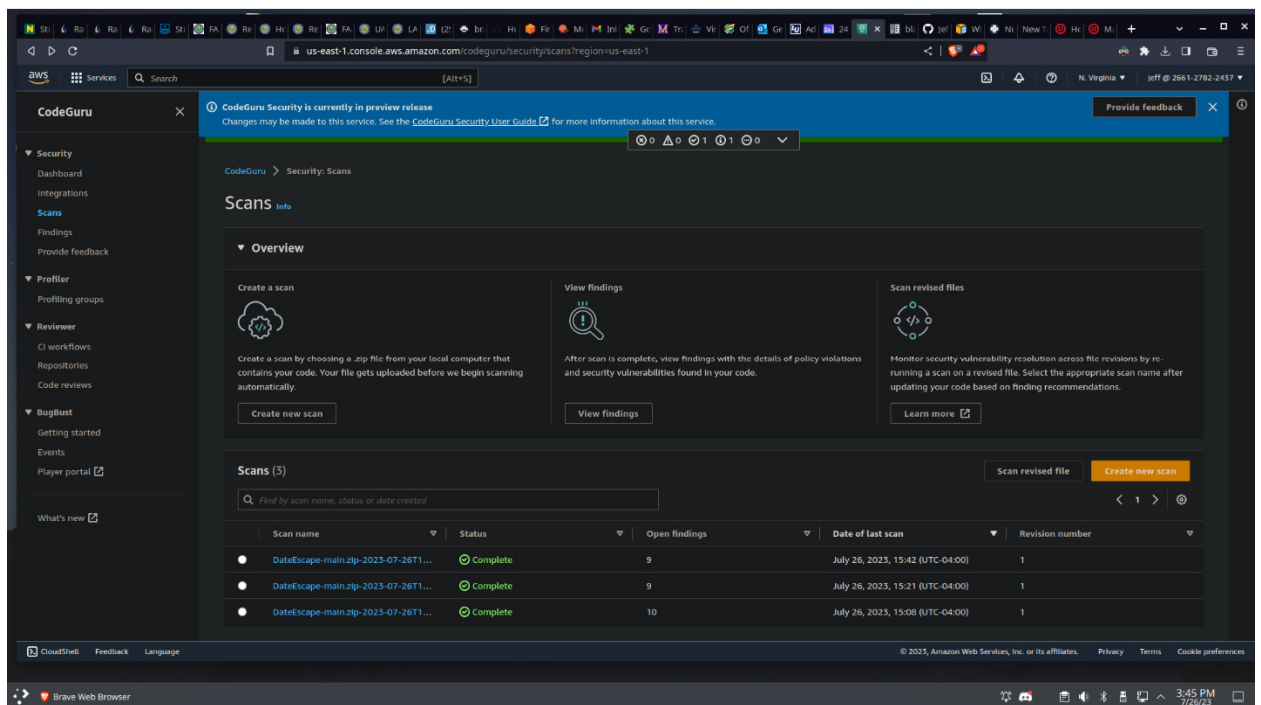
If Now type NOW: now
Calling now
An error occurred:
HTTP Error Your request was:
POST /Accounts/+18339023143/Calls.json

Twilio returned the following information:
Unable to create record: Authentication Error - Invalid username
More information may be available here:
https://www.twilio.com/docs/errors/20083

ubuntu@ip-172-31-30-115:~/DateEscape$ rm auth.token.txt
ubuntu@ip-172-31-30-115:~/DateEscape$ rm my.phone.txt
ubuntu@ip-172-31-30-115:~/DateEscape$ rm twilio.phone.txt
ubuntu@ip-172-31-30-115:~/DateEscape$ ls
README.md account_sid.txt assets.classic.link.txt initial-setup.py make-a-call.py requirements.txt test-connection.py test.PY
ubuntu@ip-172-31-30-115:~/DateEscape$ rm assets.classic.link.txt
ubuntu@ip-172-31-30-115:~/DateEscape$ rm account_sid.txt
ubuntu@ip-172-31-30-115:~/DateEscape$ python3 initial-setup.py
python3: can't open file '/home/ubuntu/DateEscape/initial-setup.py': [Errno 2] No such file or directory
ubuntu@ip-172-31-30-115:~/DateEscape$ ls
README.md initial-setup.py make-a-call.py requirements.txt test-connection.py test.PY
ubuntu@ip-172-31-30-115:~/DateEscape$ python3 initial-setup.py
python3: can't open file '/home/ubuntu/DateEscape/initial-setup.py': [Errno 2] No such file or directory
ubuntu@ip-172-31-30-115:~/DateEscape$ python3 initial-setup.py
Welcome to DateEscape!
Please enter your info EXACTLY or it will not work
This script will save your info to files
If you restart your computer, you can load your info from the files
You can find your Twilio Account SID and Auth Token at https://www.twilio.com/console
You can find your Twilio phone number at https://www.twilio.com/console/phone-numbers/incoming
The developer recommends copying and pasting in most cases to avoid frustration
Enter the phone number you would like to call in this format (USA country code is +1): +1xxxxxxxxx +12024597238
Enter your Twilio phone number with the country code in this format (USA country code is +1): +1xxxxxxxxx +18339023143
Enter your Twilio Account Sid: AC47ef0f1636d3145f6d8783b0ef9721e
Enter your Twilio Auth Token (remember to keep this secret): I-0ab337653b717d4ea (DateEscape)
PublicIPs: 54.91.153.75 PrivateIPs: 172.31.30.115

CloudShell Feedback Language
© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
main 0
Plasma Desktop
You, last week Ln 98, Col 26 Spaces: 4 UTF-8 LF Python 3.11.2 (venv: venv) Go Live 7:02 PM 7/26/23
```

Running the initial-steup.py Script on my Ubuntu EC2 Server. Via EC2 Connect



CodeGuru Security is currently in preview release

Changes may be made to this service. See the [CodeGuru Security User Guide](#) for more information about this service.

Provide feedback

Security

Dashboard

Integrations

Scans

Findings

Provide feedback

Profiler

Profiling groups

Reviewer

CI workflows

Repositories

Code reviews

BugBust

Getting started

Events

Player portal

What's new

Scans

Overview

Create a scan

Create a scan by choosing a zip file from your local computer that contains your code. Your file gets uploaded before we begin scanning automatically.

Create new scan

View findings

After scan is complete, view findings with the details of policy violations and security vulnerabilities found in your code.

View findings

Scan revised files

Monitor security vulnerability resolution across file revisions by re-running a scan on a revised file. Select the appropriate scan name after updating your code based on finding recommendations.

Learn more

Scans (3)

Find by scan name, status or date created

Scan name	Status	Open findings	Date of last scan	Revision number
DateEscape-main.zip-2023-07-26T1...	Complete	9	July 26, 2023, 15:42 (UTC-04:00)	1
DateEscape-main.zip-2023-07-26T1...	Complete	9	July 26, 2023, 15:21 (UTC-04:00)	1
DateEscape-main.zip-2023-07-26T1...	Complete	10	July 26, 2023, 15:08 (UTC-04:00)	1

Scan revised file Create new scan

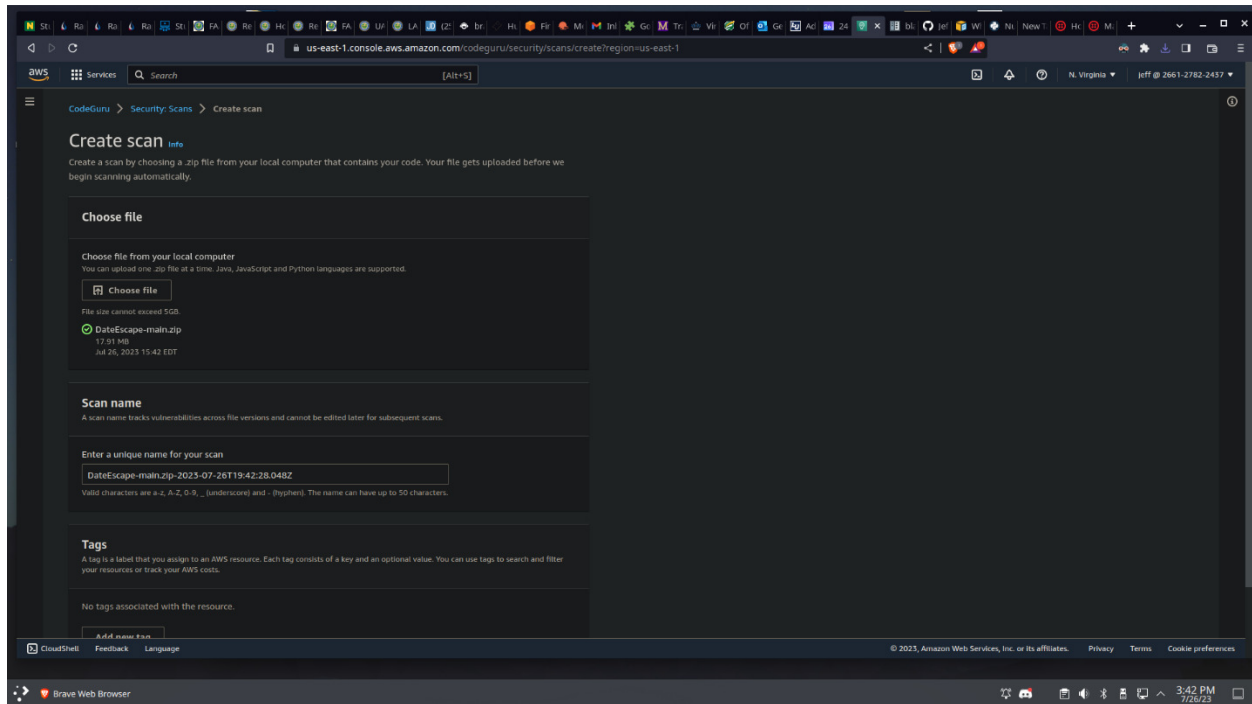
CloudShell Feedback Language

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

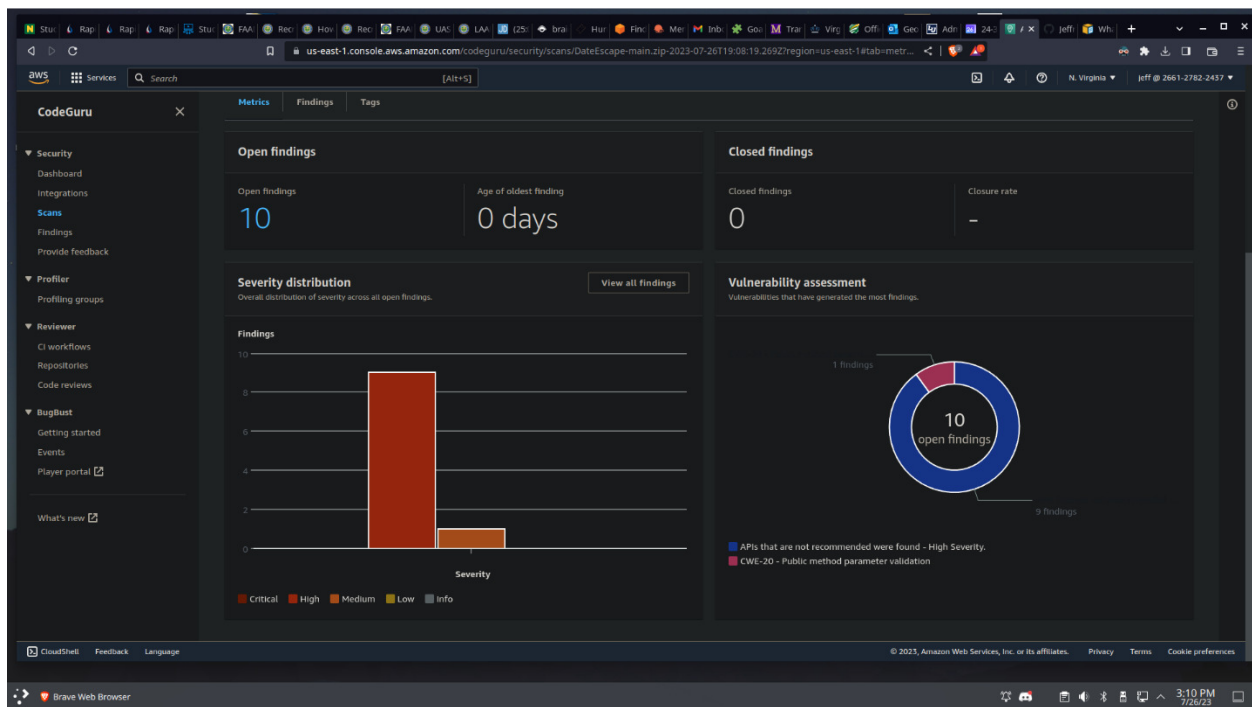
Brave Web Browser

3:45 PM 7/26/23

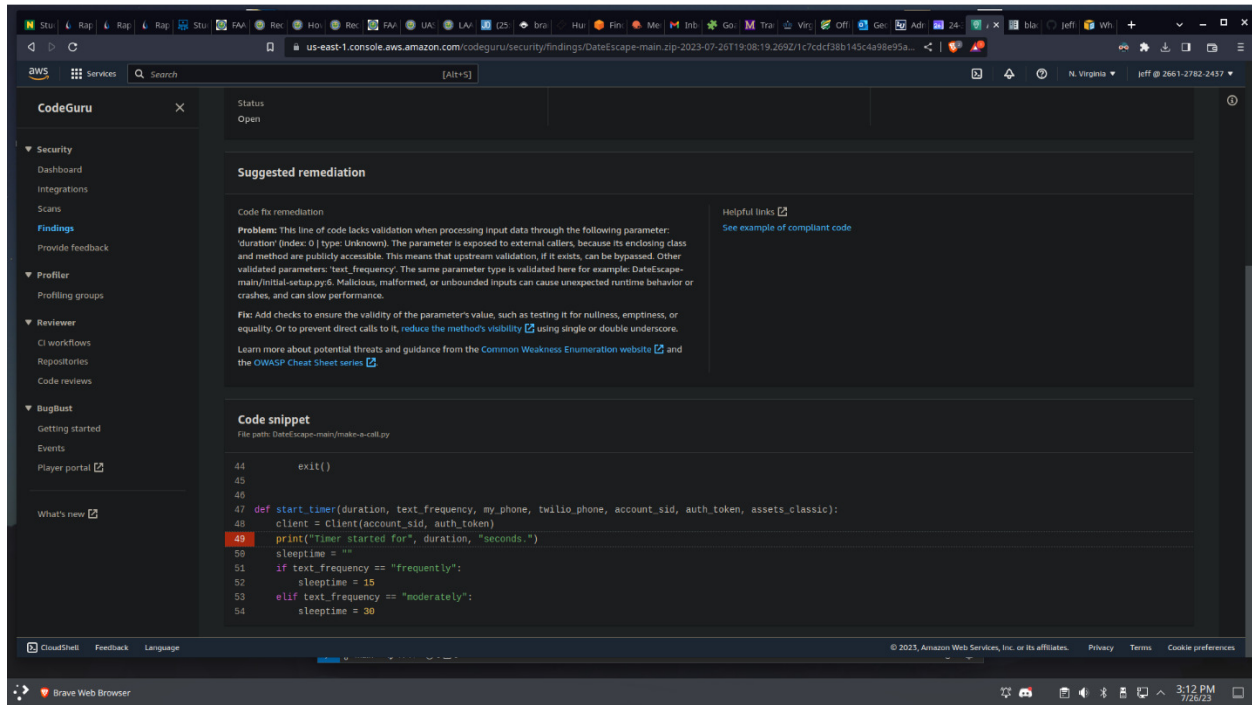
All 3 scans in CodeGuru Security.



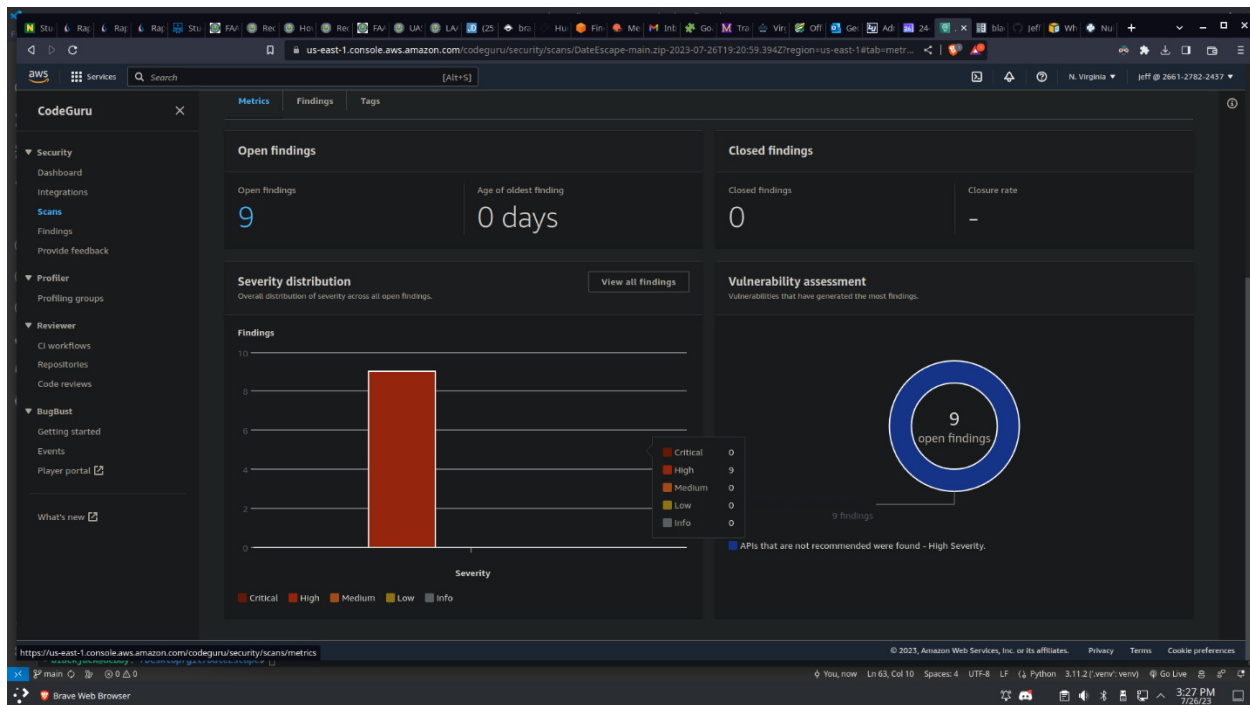
Creating a scan



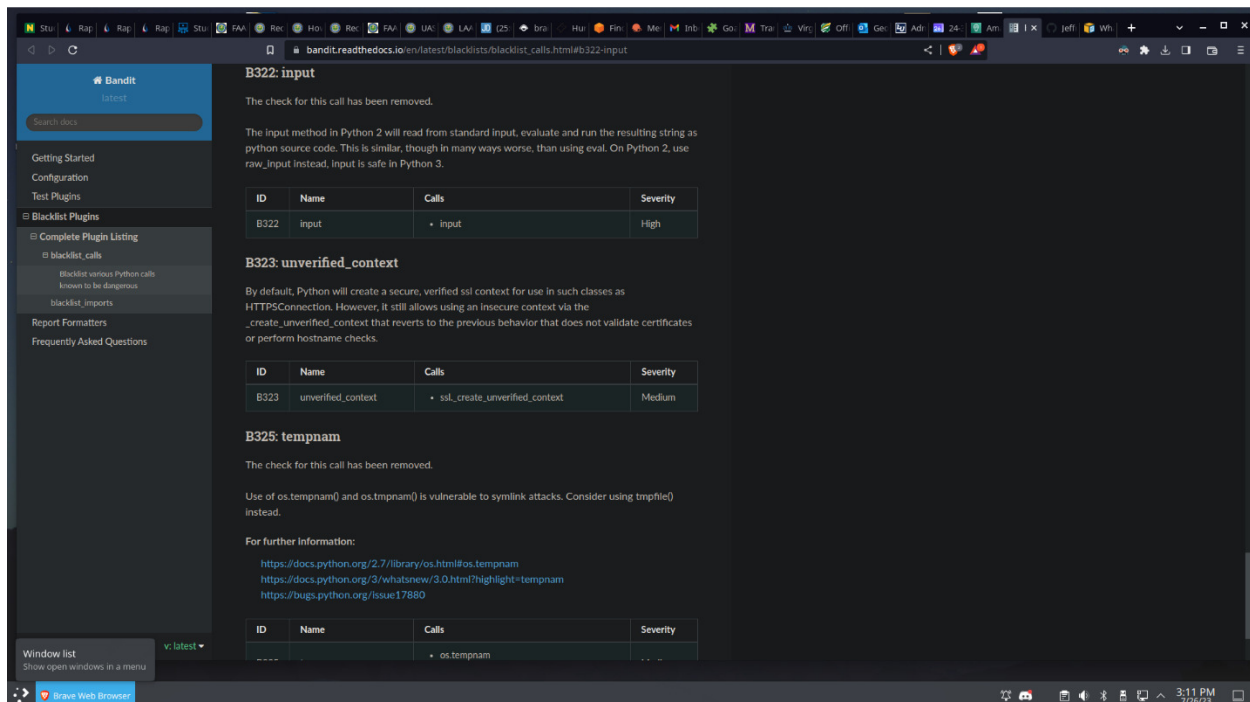
First scan results, nine red/orange are false positive, orange one is variable type checking as seen in next picture.



The type check error



After I fixed the true vulnerability, these are false positive that are only a issue with Python 2



Information about the false positive, it applies to Python 2

The screenshot displays the AWS CodeGuru console interface. The left sidebar contains navigation links for Security, Profiler, and Reviewer. The main content area shows the 'Findings' tab for a specific scan. The scan status is 'Complete'. The findings table lists 9 matches, all of which are 'High' severity vulnerabilities related to 'APIs that are not recommended were ...'. The findings were detected on July 26, 2023, at 15:21 (UTC-04:00).

Number of revisions: 1

Date of last completed scan: July 26, 2023, 15:22 (UTC-04:00)

Scan status: Complete

Scan ARN: arn:aws:codeguru-security:us-east-1:266127822457:scans/DateEscape-main.zip-2023-07-26T19:20:59.394Z

Findings (9)

Download findings

Filter findings by vulnerability or scan name: All severities: Open status: Filter by a date range: 9 matches

Vulnerability	Severity	Status	Age (days)	Time detected
APIs that are not recommended were ...	High	Open	0	July 26, 2023, 15:21 (UTC-04:00)
APIs that are not recommended were ...	High	Open	0	July 26, 2023, 15:21 (UTC-04:00)
APIs that are not recommended were ...	High	Open	0	July 26, 2023, 15:21 (UTC-04:00)
APIs that are not recommended were ...	High	Open	0	July 26, 2023, 15:21 (UTC-04:00)
APIs that are not recommended were ...	High	Open	0	July 26, 2023, 15:21 (UTC-04:00)
APIs that are not recommended were ...	High	Open	0	July 26, 2023, 15:21 (UTC-04:00)
APIs that are not recommended were ...	High	Open	0	July 26, 2023, 15:21 (UTC-04:00)
APIs that are not recommended were ...	High	Open	0	July 26, 2023, 15:21 (UTC-04:00)
APIs that are not recommended were ...	High	Open	0	July 26, 2023, 15:21 (UTC-04:00)

After resolving error on third scan.