Automate cybersecurity tasks with Python

Automation is a key concern in the security profession.
For example, it would be difficult to monitor each individual attempt to access the system.
For this reason, it's helpful to automate the security controls put in place to keep malicious actors out of the system.
And it's also helpful to automate the detection of unusual activity.
Python is great for automation.
Let's explore three specific examples of this.
First, imagine you're a security analyst for a health care company that stores confidential patient records in a database server.
Your company wants to implement additional controls to protect this information.
In order to enhance the security of the records, you decide to implement a timeout policy that locks out a user if they spent more than three minutes logging into the database.
This is because it's possible that if a user is spending too much time, it could be that they are guessing the password.
To do this, you can use Python to identify when a user has entered a username and start tracking the time until this user enters the correct password.
Now, let's cover a different example.
This time, imagine you are a security analyst working at a law firm.
There have recently been some ongoing security attacks where threat actors hack into employee accounts and attempt to steal client information.
They then threaten to use this maliciously.
So the security team is working to target all security vulnerabilities that allow these attackers to break into the company's databases.
You personally are responsible for tracking all user logins by checking their login timestamp, IP address, and location of login.
For example, if a user logs in during the early hours of the morning, they should be flagged.
Also, if they are logging in from a location that's not one of the two established work zones, you must flag their account.
Finally, if a user is simultaneously logged in from two different IP addresses, you must flag their account.
Python can help you keep track of and analyze all of this different login information.
Let's consider one final example.
Imagine you are a security analyst working at a large organization.
Recently, this organization has increased security measures to make sure all customer-facing applications are better protected.
Since there is a password to access these applications, they want to monitor all password login attempts for suspicious activity.
One sign of suspicious activity is having several failed login attempts within a short amount of time.

You need to flag users if they had more than three login failures in the last 30 minutes.
One way you could do this in Python is by parsing a static txt log file with all user login attempts to each machine.
Python could structure the information in this file, including the username, IP address, timestamp, and login status.
It could then use conditionals to determine if a user needs to be flagged.
These are just a few examples of how a security analyst might apply Python in their day-to-day work.
I hope you are as excited as I am to create solutions for security problems.

Revision #1
Created 2023-12-28 13:15:14 UTC by naruzkurai
Updated 2023-12-28 13:15:23 UTC by naruzkurai