Christian Trombley
BootCon Final Project

PhishNet: A Python-Based Script for Malicious URLs In Email

Why Choose This Project?

1. Advancement in LLMs and AI are increasing volume and sophistication of phishing attempts

2. Automated tasks allow security professionals to focus on higher priority issues, or projects

3. I wanted to work on a project that could help get me started in coding/scripting

Technical concepts applied:

- IMAP
- API use from VirusTotal
- Python scripting

Tools for the job:

- Python, chosen for its popularity and widespread use
- imap-tools library to access Gmail through IMAP
- VirusTotal API for URL analysis

Setup:

- **Used 3 Gmail accounts to simulate a defender, attacker, and a neutral party with 3 “good” links and 1 “bad”**
- **A script was then used to automatically flag bad emails in a report text file**

Difficulties:

- Gmail requires APP passwords, so the default login will not work
- Had to resolve an error caused by an emoji being present in the body/header of an email
- Had to craft a few different links to ensure clear demonstration of the script
- VirusTotal’s Free API is rate limited. Which was one thing that caused the scope of this project to be simplified

DEMO

Show Email Inbox -> Run Script -> Show Text Report

PhishNet
Summary and Future
Expansion

What just happened?

- The script logged into the defined email using imap
- It then pulled URLs from the top 4 unread emails
- These URLs were then run against the VirusTotal API
- The scan results are then pulled from VirusTotal, and the flagged Emails are added to a report .txt file for easy viewing

This script could be expanded in a few ways for enterprise

-script could be changed to run against multiple mailboxes or an entire company mail server

-can be turned into a scheduled task to run hourly/daily/weekly etc.

-Report results could be formatted and set up to feed into SIEM software such as Splunk

-Set up a notification to analysts when flagged emails are found