Sprinkler: A Multi-Service Password Sprayer

Sam Ederington, Prompt Eua-anant, advised by Jeff Ondich

QR

What's password spraying?

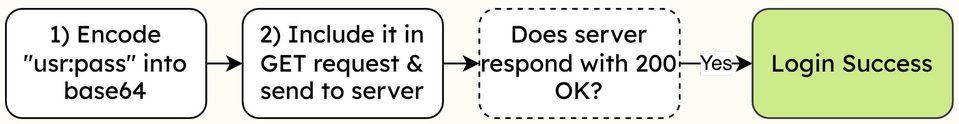
- Password spraying is trying to login to a target server via multiple user accounts with a list of passwords as quickly as possible.
- Password spraying is used for hacking and security purposes, finding accounts with vulnerable login permissions to attack or require changing.
- Password spraying simulates a user's login request to whatever type of service they would use (website, remote desktop, …), sending a username and password and checking for a successful login before terminating the connection.

Our goal

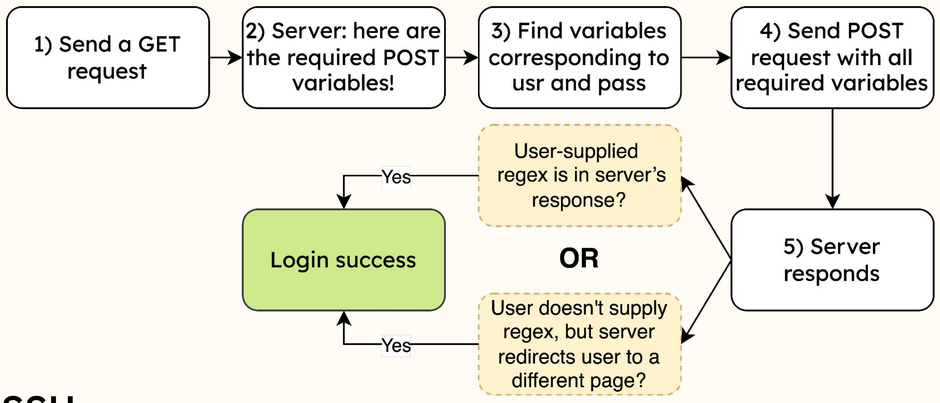
- Make a fast password sprayer that works for ssh, http-get, http-post
- Make a test server and machine compatible with the services we need

How sprinkler works for each service?

HTTP-Get (basic auth)



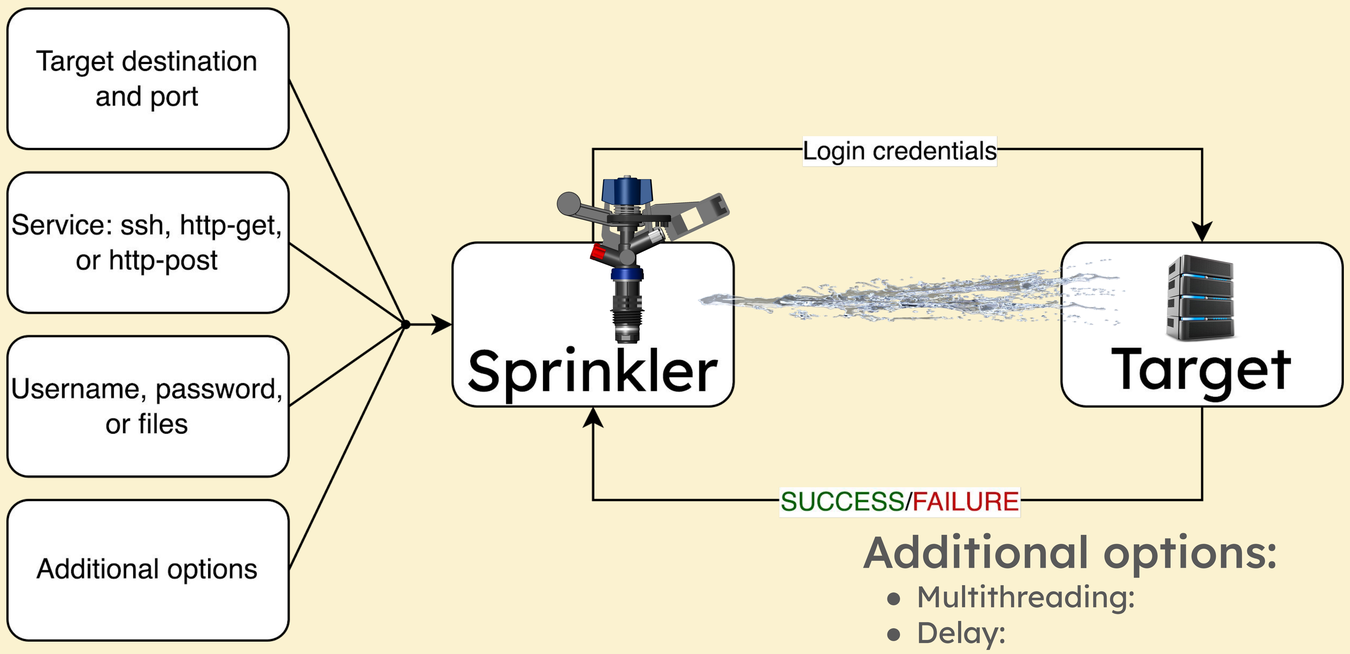
HTTP-Post



SSH

1. We used libssh library functions to connect to the server
2. Format and send the login request to the server
3. Login success = server responds with authentication successful and gives access to its terminal

Visualization of Password Spraying



Additional options:

- Multithreading:
- Delay:
- Regex for success condition:

The Process:

- Sprinkler is given a list of inputs, including a username or username file, passwords, and a target destination as well as any options the user wants to enable
- Sprinkler reformats the inputs into login attempts and sends them off
- As soon as the target sends back a successful login message, Sprinkler marks that attempt as a match for the user and stops spraying to that user
- Continues spraying until every user has had a login or every supplied password fails
- Sprinkler prints out the valid logins

Optimizations for speed

- Multithreading, or using different parts of the processor to perform multiple attempts simultaneously
- Working directly with computer memory, which lets us choose what information we need on hand to perform actions faster

Acknowledgements: Jeff Ondich, knowledgable users on StackOverflow, and libssh