System Administration & Security

COMP 175 | Fall 2021 | University of the Pacific | Jeff Shafer

Lab 8 Discussion: Scripting (with DNS, Log Files, AWS CLI)

Lab 8 – Scripting

Objectives

- Automatic DNS Updates
- Log file analysis (w/scripting)
- SSH Connection Limiting

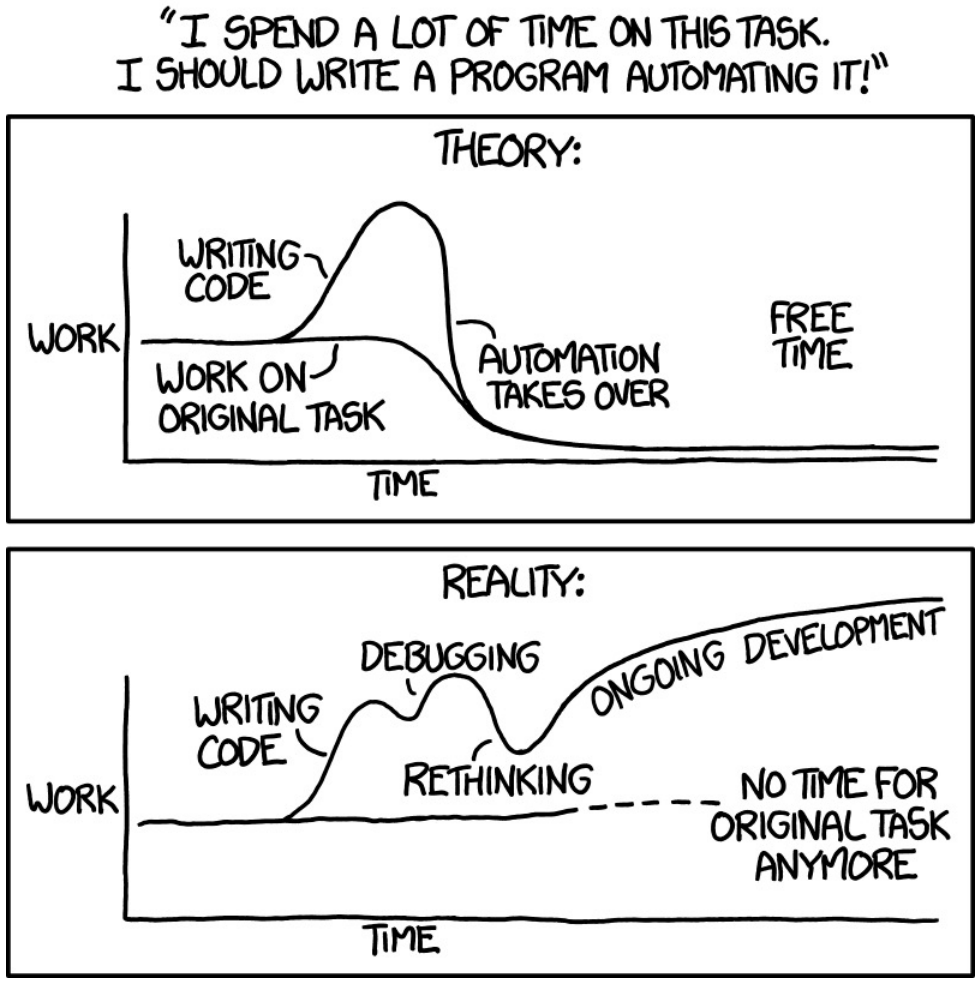
Discussion

- DNS
- AWS CLI
- Fail2Ban

AWS CLI

DNS Motivation

- We want some nice DNS entries for tigerenterprise
  - `www.STUDENT-NAME.tigerenterprises.org`
  - `vpn.STUDENT-NAME.tigerenterprises.org`
  - And others in the future?
- But we're too *cheap* to pay for an ElasticIP to have a 100% reserved IP address…
- … and we're too *lazy* to update Route53 every time we launch a new instance



Let's automate this menial task!

https://xkcd.com/1319/

AWS Command Line Interface

- Free, cross-platform (Windows, MacOS, Linux)
  - https://aws.amazon.com/cli/
- Intended to provide equivalent functionality to the browser-based AWS Management Console, but at the command-line
  - Scriptable!
  - *More capable in some instances (not every obscure action can be done via the web console)*

AWS Command Line Interface

- Traditional "getting started" method of using AWS CLI is to load it with your user credentials
  - AWS Access Key ID
  - AWS Secret Access Key
  - *AWS Academy users can find this in your Vocareum portal page – click on "Info" and then "show AWS CLI"*
    - *Note that your credentials expire after 4 hour…*
- Can also grant permission to specific instances to run specific AWS CLI operations, and thus no additional authentication is required

DNS Update

- In a Bash script, run at startup:
  1. Load some variables with desired DNS name and Route53 Hosted Zone ID
  2. Query Amazon (via a special-purpose URL) about your current public IP
  3. Request Amazon (via CLI) change a Route53 resource record to reflect current public IP address

Curl

- `curl` is a tool to transfer data from or to a server
- Supports myriad protocols!
  - HTTP, HTTPS, SCP, SFTP, etc…
- Supports myriad features
  - Proxies, user authentication, cookies, file transfer resume, etc…

```
$ MY_IP=$(curl -s http://169.254.169.254/latest/meta-data/public-ipv4/)
$ echo ${MY_IP}
54.148.163.48
```

Fail2Ban

Fail2Ban

- Python program intended to *slow down* automated password-guessing attacks against servers
- Monitors log files for patterns that indicate attack (e.g. repeated failed logins from an IP) and takes action (e.g. update firewall to ban IP temporarily)
- Many applications supported
  - Apache/Nginx, SSHD, qmail, proftpd, …
- Limited effectiveness against a distributed brute-force attack

Wrap-Up

- Questions?
- Concerns?

- **Today**
  - **Lab 7** – VPN
  - **Lab 8** – Scripting