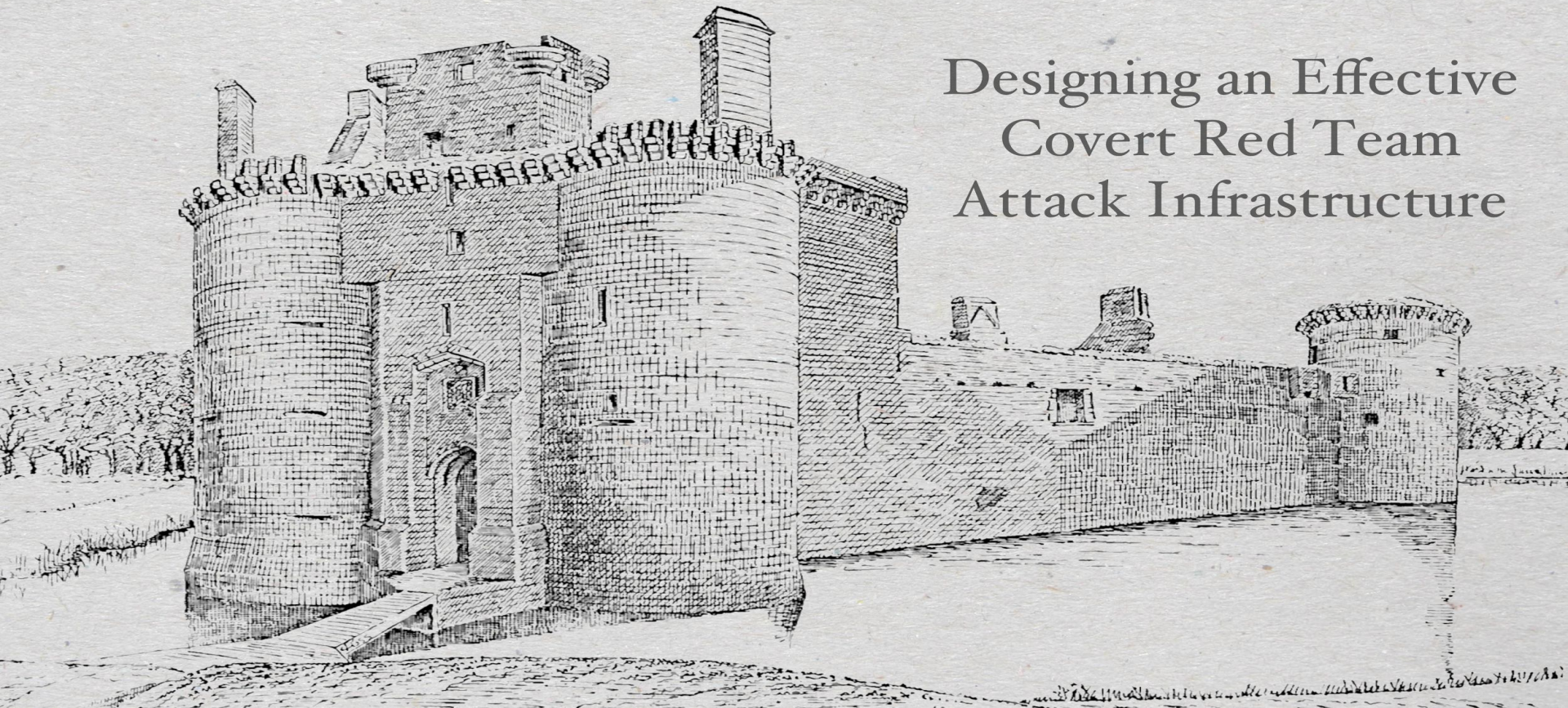
Building a Better Moat

Designing an Effective Covert Red Team Attack Infrastructure



whoami

Jeff Dimmock (@bluscreenofjeff)

- Adversary Simulation Lead at SpecterOps
  - Adaptive Threat Division alumni
  - Avid blogger (https://bluescreenofjeff.com/)
  - Writes some Aggressor scripts
- Speaker at BSides NoVA 2017 and HackMiami
- Participated in Pacific Rim CCDC since 2014
- Maintains the Red Team Infrastructure Wiki w/ @424f424f (http://github.com/bluscreenofjeff/Red-Team-Infrastructure-Wiki)

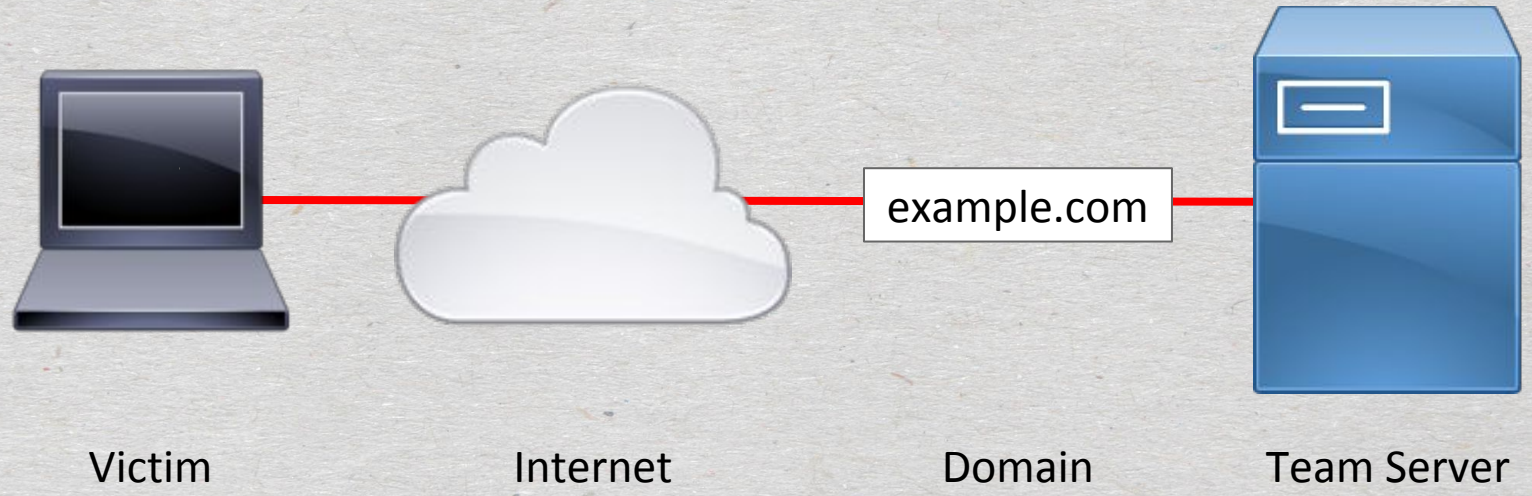
bsoj.co/ArcticCon2017

Agenda

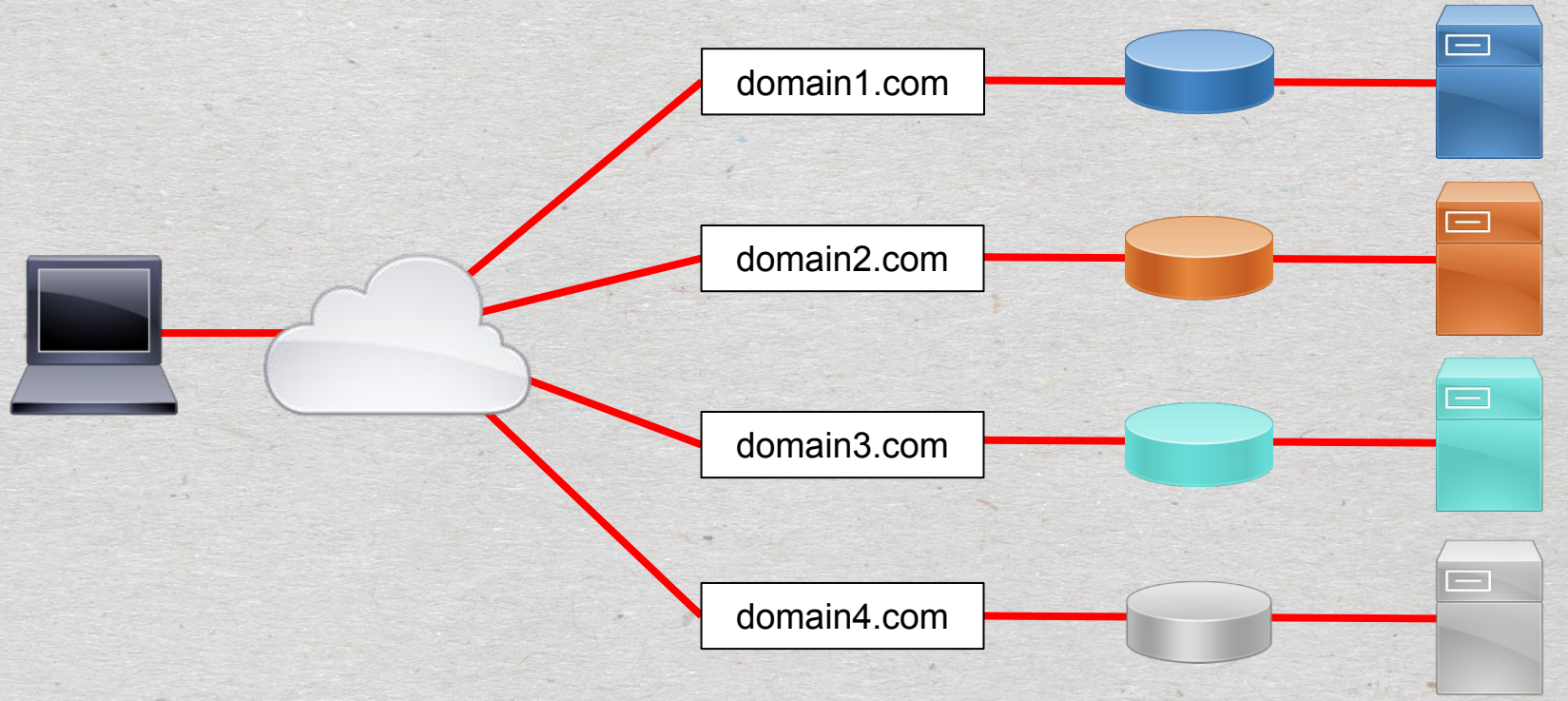
- Introduction
- Design Considerations
  - General
  - Domains
  - Payloads
  - C2
  - SMTP
- Example Infrastructure Designs
- Key Takeaways

What is a Covert Red Team Attack Infrastructure?

Standard Pentest Setup



Covert Red Team Attack Infrastructure



Benefits of a Covert Infrastructure

- Blends in with non-malicious infrastructure
- Builds in defenses against Blue team
- Resilience when an asset is blocked
- Flexibility to adapt as the test progresses

Design Considerations

Design Considerations - General

- Segregate all assets based on function
- Place a redirector in front of every asset
- Blend in with normal traffic for the target
- Configure assets in a way that individual pieces can be rolled
- Keep the length of the test in mind when designing

Design Considerations - General

- Don’t go overboard for your target
- Tailor the design for anticipated or encountered obstacles
- Evaluate infrastructure primitives before choosing one
- Be fluid as the test goes; redesign as needed

(Sidenote: document EVERYTHING)

Design Considerations - Domains

- Always use categorized domains*
- Pre-categorized vs. home-grown
- Matching the target's domain categorization can help blend in
- Alternatively, organizations often have domain category whitelists (try Healthcare/Finance!)

*unless emulating an actor who doesn't

Design Considerations - Domains

- Domains should match planned pretext
- Good choices:
  - Impersonate target company
  - Similar to popular companies/services (Google, Microsoft, etc.)*
  - Generalized domains that match industry

*There are precedents for companies coming after squatters

Design Considerations - Payloads

- Payload type
  - HTA, LNK, PowerShell download cradle, etc
- Delivery method (attached, web-based)
  - Attachment
  - Download link (server vs. cloud storage)
- Access control options
  - IP/fingerprint-based restrictions
  - Expiring payload links

Design Considerations - Payload Redirection

- “Dumb pipe”
  - socat/iptables
  - Forwards all connections
  - No monitoring of requests
- Filtering
  - Apache mod_rewrite, nginx
  - Perform conditional processing on requests
  - Modify rulesets on the fly

Design Considerations - Command and Control (C2)

- Longhaul
  - Used to regain access
  - Persistence
  - Slow check-ins
- Shorthaul
  - Used for primary operating
  - Burned frequently
  - Faster check-ins
- Number of servers

Design Considerations - Command and Control (C2)

- Popular protocol choices:
  - HTTP(S)
  - DNS
  - Domain Fronting (over HTTPS)
  - Third-party

Design Considerations - Command and Control (C2)

Attribute	**HTTP(S)**	**DNS**	**Domain Fronting**	**Third-Party**
Latency	Low	High	Medium	Medium
Likelihood to Work	Average	High	High	High
Detectability	Average	High	Low	Low
Ease of Blocking	Average	Low	Low	Low
Ease of Setup	Easiest	Easy	Medium	Medium/Hard

Design Considerations - Command and Control (C2)

- Traffic shaping:
  - Custom network footprint
  - Emulate a specific application or threat actor
  - Modify potential signatures (URIs, user agents, etc.)
  - Modify any host implant attributes as much as possible
- Use custom profiles for each server

Design Considerations - Command and Control (C2)

- Choose an appropriate protocol and redirection method per server
  - Validate latency before launching the test
- Keep servers separate
  - Shorthaul and longhaul servers fulfill different roles
  - Be disciplined with each server type
  - Never cross the streams!
- Don’t be afraid to roll new infrastructure

Design Considerations - C2 Redirection

- “Dumb pipe”
  - Easiest setup, no filtering protections
- Filtering
  - Longer setup, higher resilience
- Domain Fronting/Third-Party C2
  - Requires setup or development
  - Blends in well
  - Very hard to block

Design Considerations - SMTP

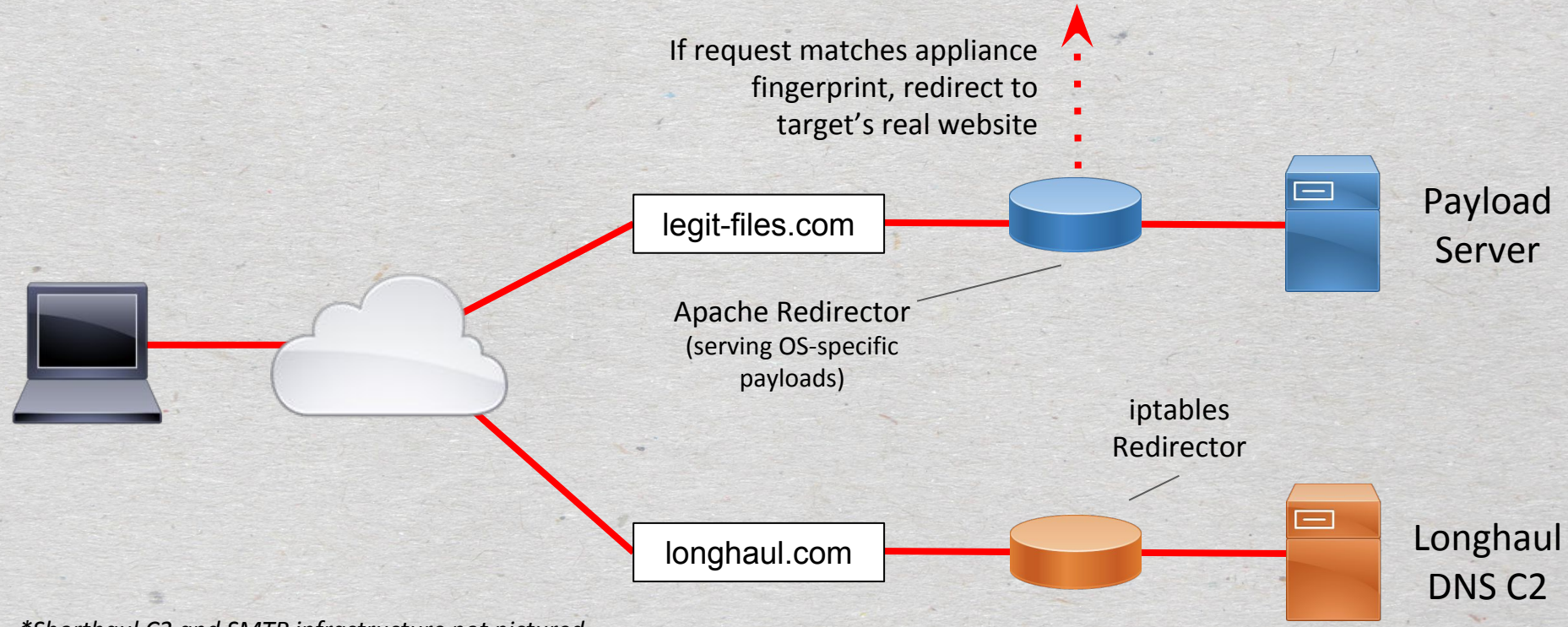
- Self-setup
  - Set up yourself
  - DKIM, SPF, PTR records needed
  - Can be time-intensive
- Third-party
  - Automatic reputation
  - Anti-spam controls
- Open Relay
  - Can use target’s own server against them
  - Anti-spam controls could get you flagged

Example Infrastructure Designs

Example Infrastructure Design #1

- Email attachments blocked
- Pre-phish revealed email appliances
- Appliances crawl all links in emails
- No intel about web browsing restrictions
- Mixed OS environment

Example Infrastructure Design #1

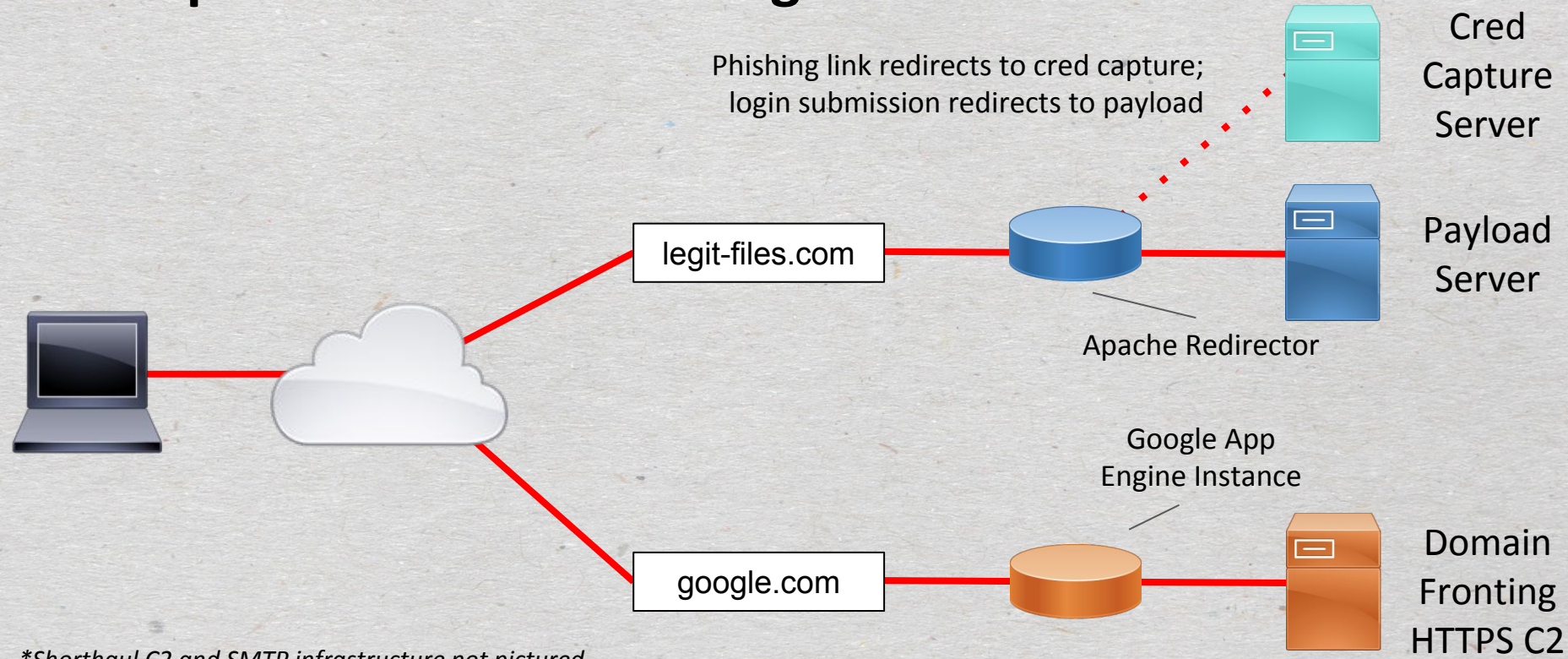


**Shorthaul C2 and SMTP infrastructure not pictured*

Example Infrastructure Design #2

- About half of users don’t have internet access
- Those with internet access are heavily restricted
- Publicly-accessible OWA and remote access logins

Example Infrastructure Design #2

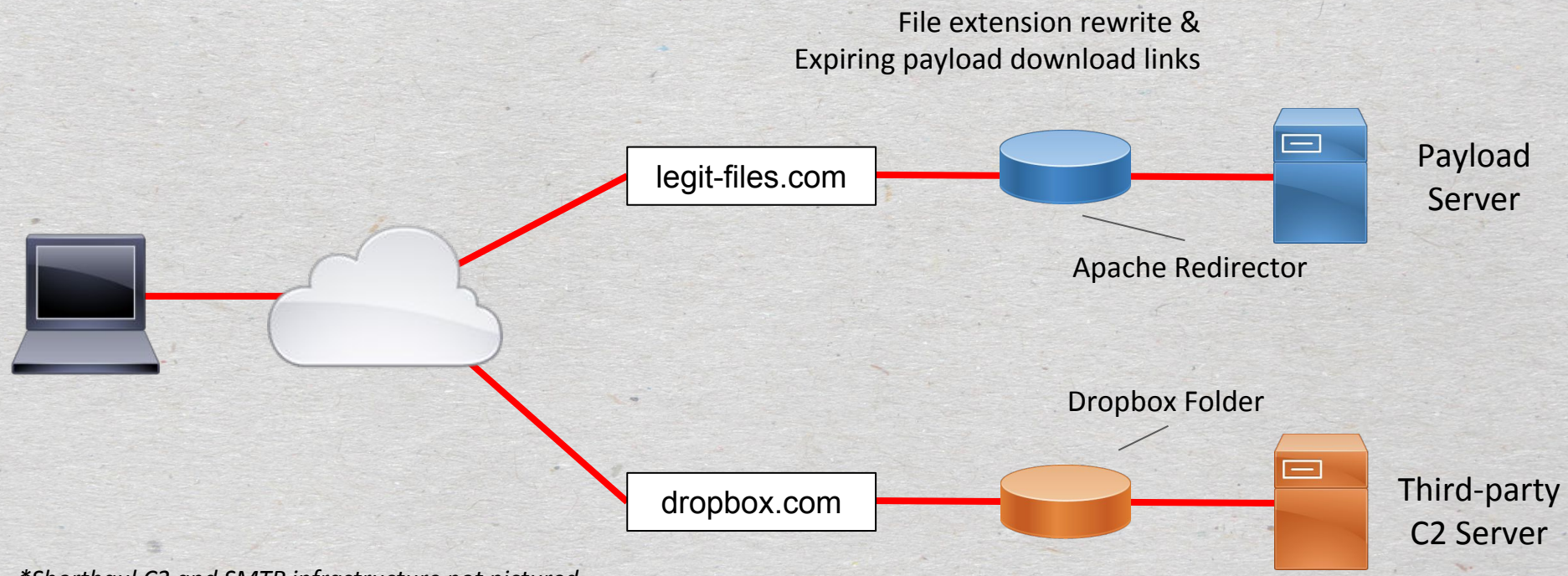


**Shorthaul C2 and SMTP infrastructure not pictured*

Example Infrastructure Design #3

- Email attachments blocked
- Email appliance reviews all links and blocks wide range of file extensions
  - docx allowed
- Pre-phishing was manually acted on quickly
- Found case study for the target's deployment of a popular online storage service

Example Infrastructure Design #3



**Shorthaul C2 and SMTP infrastructure not pictured*

Key Takeaways

- Segregate assets
- Use redirectors, ideally with filtering
- Stress test assets for operational latency
- Document EVERYTHING
- Don’t go overboard - achieve your objectives

bsoj.co/ArcticCon2017

bit.ly/RedTeamInfrastructure

specterops.io