Screaming Frogs and Crawling Bots

SIP UA Device File Security

Jon Chleboun | ECG

The Road to Fraud

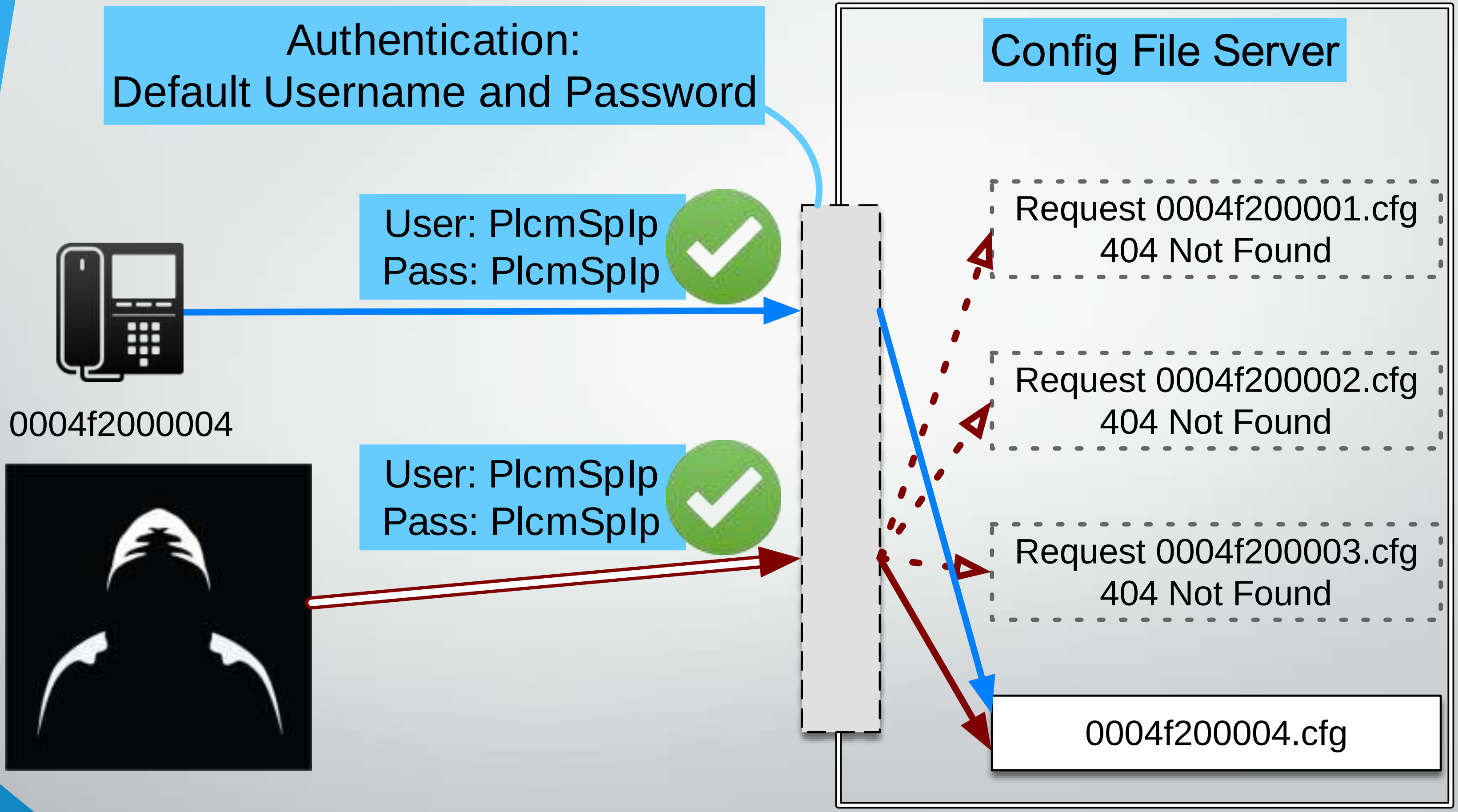
is Paved with Stolen Config Files

- “PlcmSpIp”
- “UserAgent: Screaming Frog SEO Spider/8.1”

-
  ```
  “GET /dms/SPA525G/a4934c8f67e3.xml” 404 Not Found
  “GET /dms/SPA525G/a4934c8f67e4.xml” 404 Not Found
  “GET /dms/SPA525G/a4934c8f67e5.xml” 404 Not Found
  “GET /dms/SPA525G/a4934c8f67e6.xml” 404 Not Found
  “GET /dms/SPA525G/a4934c8f67e7.xml” 200 OK
  ```
- “We have begun blocking international calls from your account due to abnormal calling patterns”

SIP UA Device File Security

- Passwords are a drag
- Cryptography saves the day
- Becoming less lazy is hard
- Challenges
- Tools



Authentication:
Default Username and Password

User: PlcmSpIp
Pass: PlcmSpIp

0004f2000004

User: PlcmSpIp
Pass: PlcmSpIp

Config File Server

Request 0004f200001.cfg
404 Not Found

Request 0004f200002.cfg
404 Not Found

Request 0004f200003.cfg
404 Not Found

0004f200004.cfg

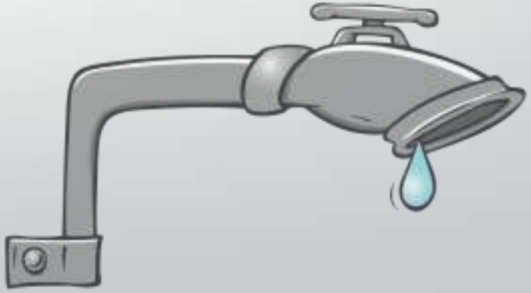
Device File Access with Default Passwords

Individual Device Passwords are a Drag (and we are lazy)

- Device config files can be secured via ye olde username and password
- Not Too Terribly Bad: A unique, good password for each phone
- Slightly Less Bad: The same decent password for all phones
- Bad: Default username and password are used
- Really Bad: No authentication

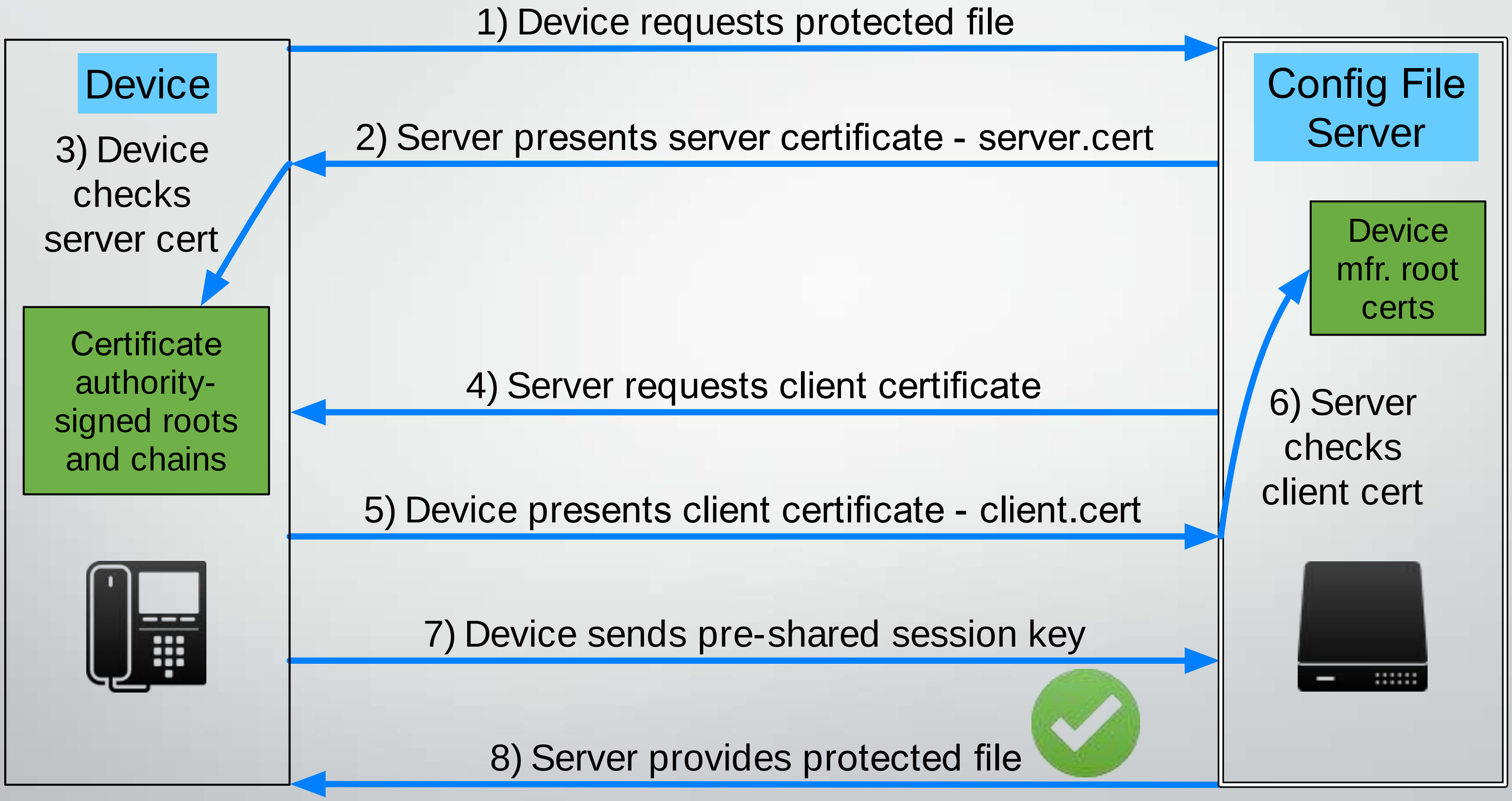
Why We Keep the Fraudsters in Business

- Unique device passwords = support nightmare
- Passwords are like old pipes
- Humans (including VoIP nerds) are lazy



Cryptography Parachutes In to Save the Day

- Modern UA devices got a client certificate at birth
- Signed by manufacturer’s root, with the device’s MAC encoded
- Web servers and load balancers can do Mutual TLS



Device

3) Device checks server cert

Certificate authority-signed roots and chains

1) Device requests protected file

2) Server presents server certificate - server.cert

4) Server requests client certificate

5) Device presents client certificate - client.cert

7) Device sends pre-shared session key

8) Server provides protected file

Config File Server

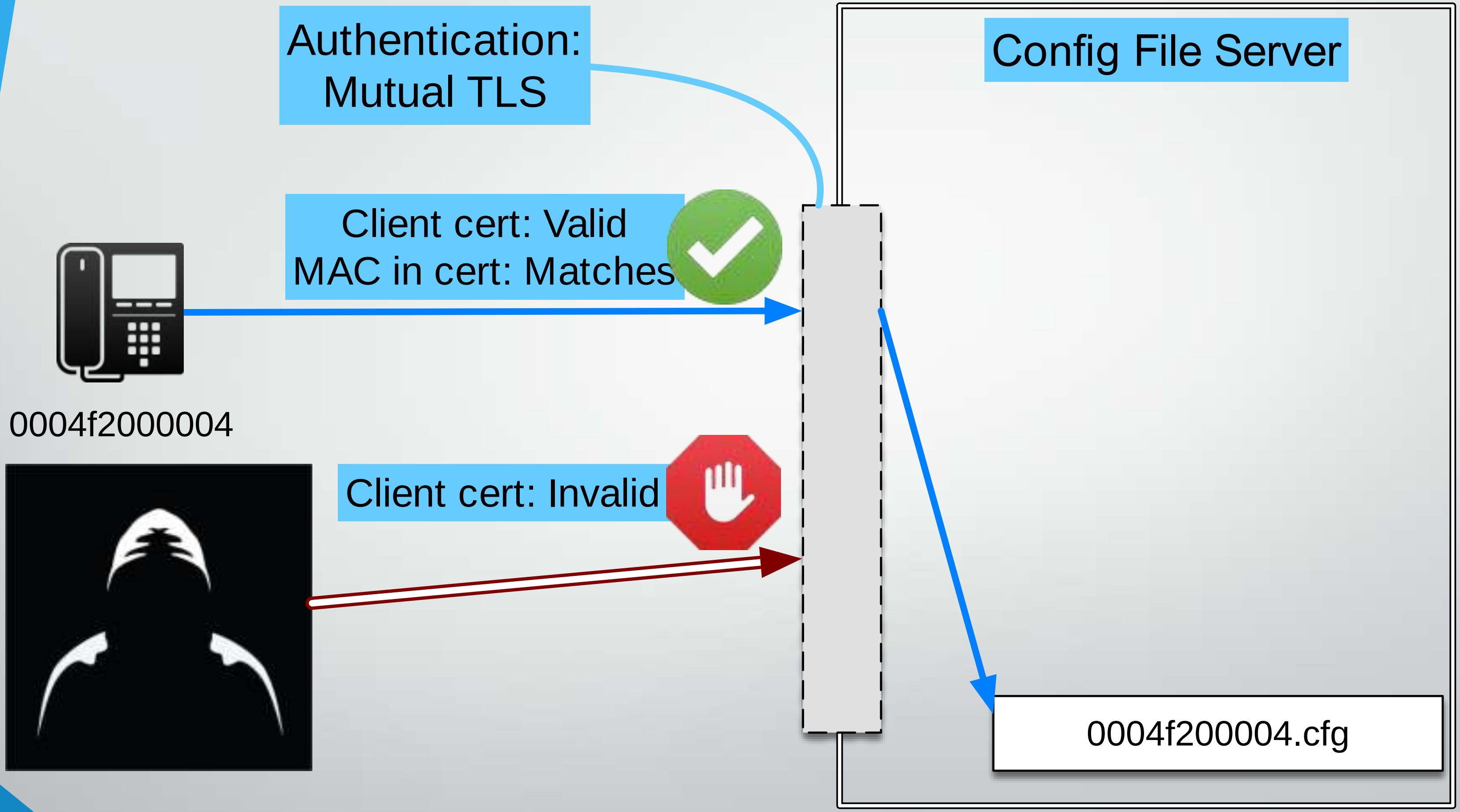
Device mfr. root certs

6) Server checks client cert

Mutual TLS Handshake

Knowledge is Power

- This device is not a Screaming Frog
- This device is a real SIP phone, from the manufacturer I expect
- This device is THE ONE DEVICE IN THE WORLD that should have this file



Authentication:
Mutual TLS

Config File Server

Client cert: Valid
MAC in cert: Matches

0004f2000004

Client cert: Invalid

0004f200004.cfg

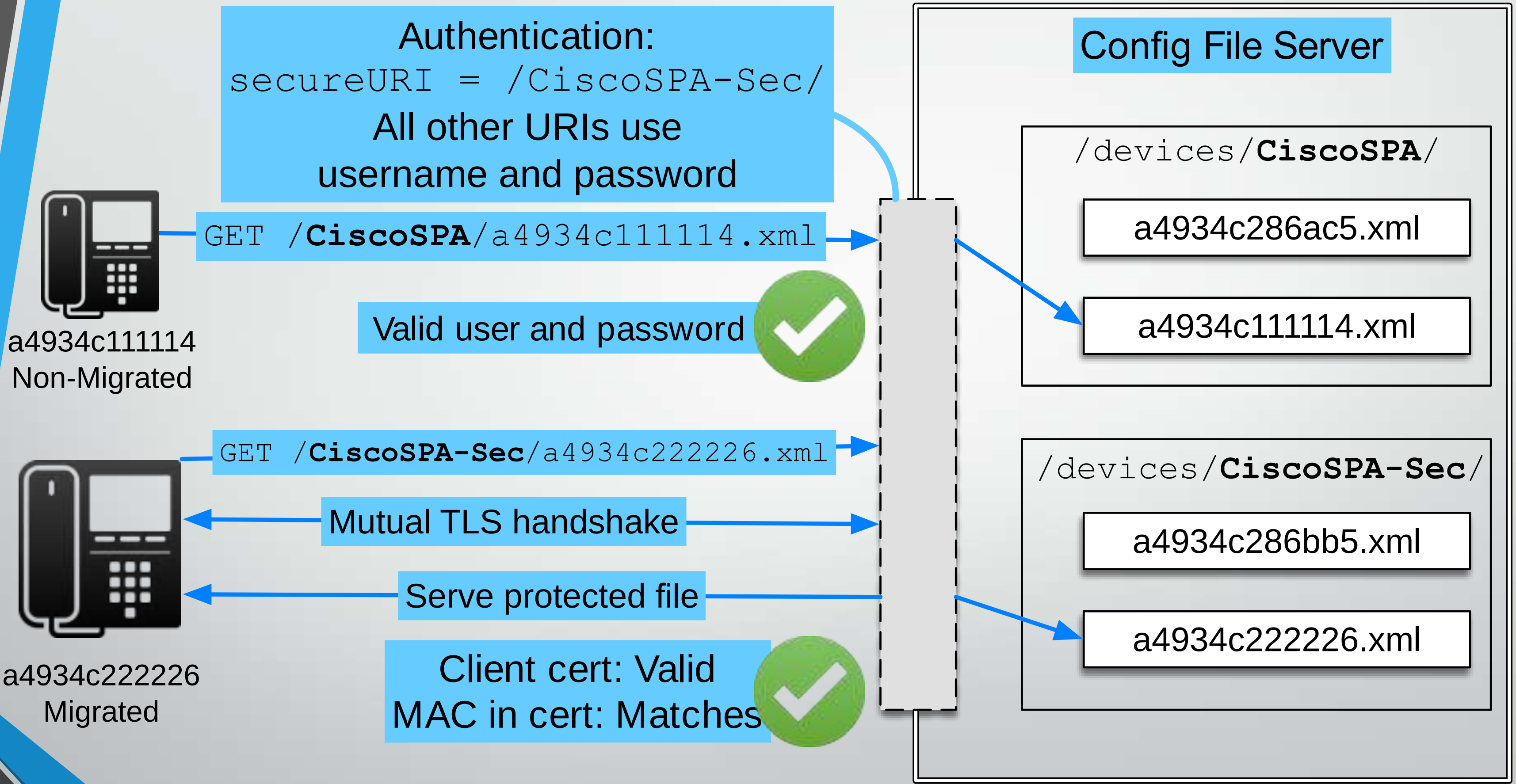
Device File Access with Mutual TLS

Becoming Less Lazy is Hard

- Greenfield is easy
- Harder when people want to use their phones while you flip the switch
- Challenges
- Tools To Minimize Pain

Challenge: Migrate Profile Class/Type

- Mutual TLS turned on per URI path (BroadWorks and BIG-IP LTM)
  - `secureUri = CiscoSPA-Sec/, Polycom-Sec/, Yealink-Sec/`
- Migration approach can limit risk
- Device Profiles rebuilt with new Profile Class/Type



Authentication:
`secureURI = /CiscoSPA-Sec/`
All other URIs use
username and password

Config File Server

/devices/**CiscoSPA**/

a4934c286ac5.xml

a4934c111114.xml

/devices/**CiscoSPA-Sec**/

a4934c286bb5.xml

a4934c222226.xml

a4934c111114
Non-Migrated

GET /**CiscoSPA**/a4934c111114.xml

Valid user and password ✓

a4934c222226
Migrated

GET /**CiscoSPA-Sec**/a4934c222226.xml

Mutual TLS handshake

Serve protected file

Client cert: Valid
MAC in cert: Matches ✓

Some URIs Secured, Some Not

Challenge: Migrate Profile Class/Type

- Tools To Minimize Pain
  - Automation
  - Streamlined Product

Challenge: Certs Factory Installed? ¯_(ツ)_/¯

- For certain models, some phones have signed certs, some don’t!
- Example: Polycom began installing certs between May 2009 and December 2011 – 2.5 year span
    - SoundPoint IP 550 – December 2009 start date
- Difficult (but possible) to check remotely

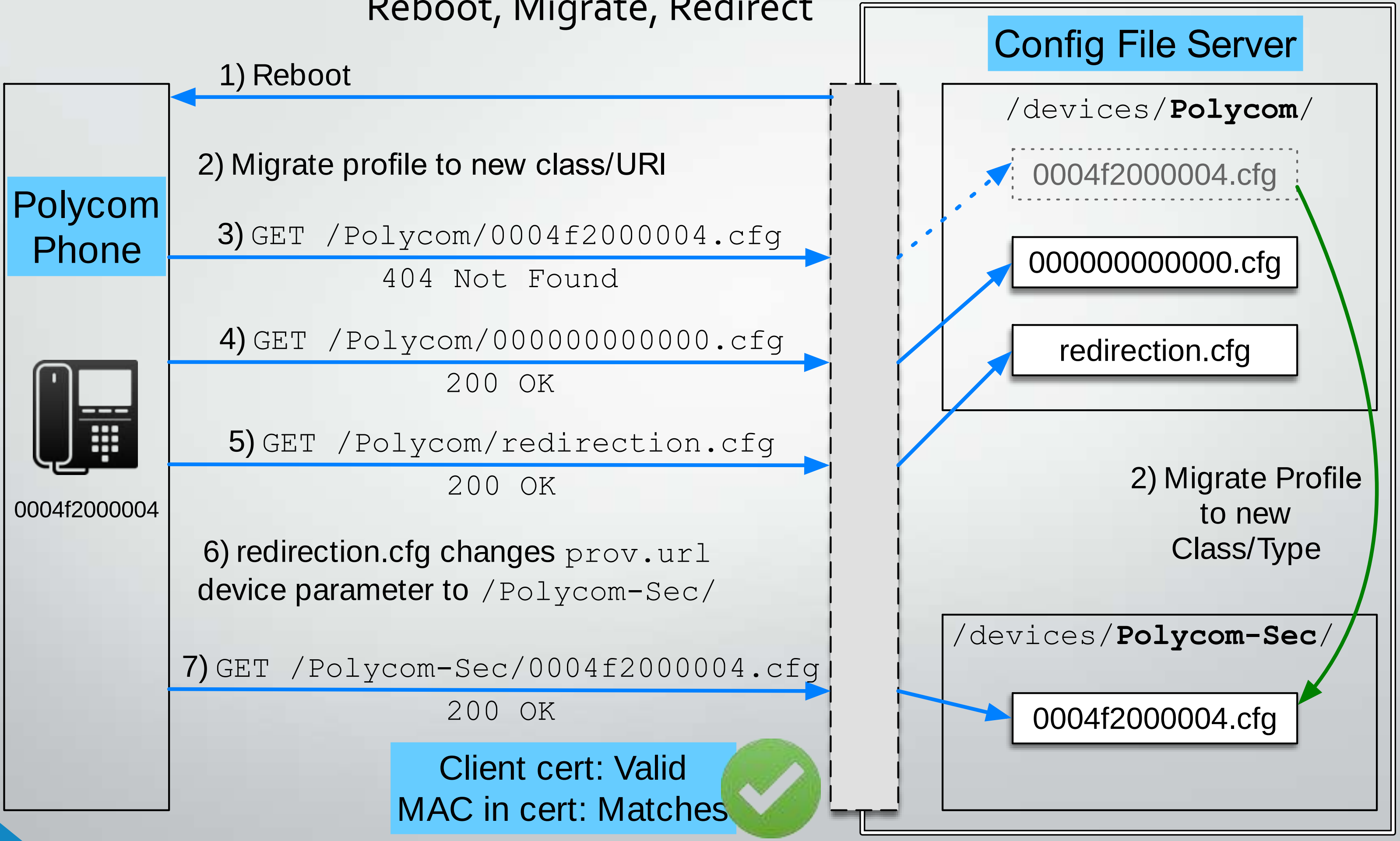
Challenge: Certs Factory Installed? ¯_(ツ)_/¯

- Tools To Minimize Pain
  - Streamlined Product
  - Product Control
  - “I give up!”
    - `secureUri = Polycom/, CiscoSPA/, Yealink/`
    - `whitelistUri = Polycom/SPIP/`

Challenge: Timing

- If you change the URI before telling the phone, it breaks
- If you tell the phone before changing the URI, it breaks
- Tools To Minimize Pain
    - Migrate in chunks
    - Reboot, Migrate, Redirect

Reboot, Migrate, Redirect



Polycom Phone

0004f2000004

1) Reboot

2) Migrate profile to new class/URI

3) `GET /Polycom/0004f2000004.cfg`

`404 Not Found`

4) `GET /Polycom/000000000000.cfg`

`200 OK`

5) `GET /Polycom/redirection.cfg`

`200 OK`

6) redirection.cfg changes `prov.url` device parameter to `/Polycom-Sec/`

7) `GET /Polycom-Sec/0004f2000004.cfg`

`200 OK`

Client cert: Valid
MAC in cert: Matches ✓

Config File Server

`/devices/`**`Polycom`**`/`

0004f2000004.cfg

000000000000.cfg

redirection.cfg

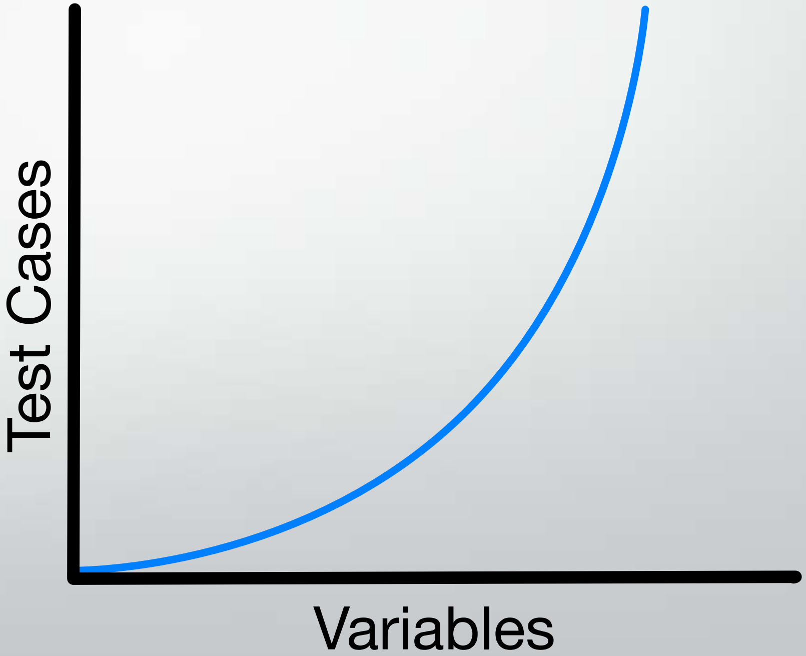
2) Migrate Profile to new Class/Type

`/devices/`**`Polycom-Sec`**`/`

0004f2000004.cfg

Challenge: Multiplication

- 11 phone models
- * 5 possible firmware versions
- * 20 test cases in your test plan
- = 1100 tests
- Tools To Minimize Pain
  - Upgrade First
  - Streamlined Product



Challenge: If the frogs don't get you, the bugs will

- Old (or buggy) client firmware may not support Mutual TLS
  - "HTTPS? Hogwash. SHA-2? In your dreams. Client certificate challenge? Fuggedaboutit"
- Tools To Minimize Pain
  - Testing, Testing, Testing
  - Upgrade First
  - Upgrade First Workaround
    - `secureUri = CiscoSPA-Sec/, Yealink-Sec/, Polycom-Sec/`
    - `whitelistSuffix = 000000000000.cfg, *.bin, *.sip.ld, *.jpg`

Toolkit

- Streamlined Product
  - Minimize number of firmware versions and Profile Classes/Types
  - Move out of the Dark Ages (stop supporting models that can’t be made secure)
- Upgrade First
- `whitelistUri = Polycom/SPIP/`
- `whitelistSuffix = 000000000000.cfg, *.bin, *.sip.ld`
- Tight Product Control

Toolkit continued

- Reboot, Migrate, Redirect
- Migrate in chunks
- Automation
- Testing, Testing, Testing

Three Cheers for Becoming Less Lazy

The End