Android weblogin: Google's Skeleton Key

Craig Young, Tripwire VERT

whoami

I research, identify, and disclose vulnerabilities as a senior researcher in Tripwire VERT.

I enjoy long bike rides, breaking things which fail to sanitize input, and building furniture with my wife on the weekend.

DISCLAIMER: I am definitely not an Android developer.

Talk Overview (tl;dr)

1. Android trades security for convenience
2. *weblogin:* can bypass password prompts
3. Security tools do not detect token egress
4. 1 token can fully compromise Google Apps

About *weblogin:*

- Android Token Type:

weblogin:service=youtube&continue=https://www.youtube.com/

- Grants cookies for the desired service

- Acts in lieu of password entry

Abusing *weblogin:*

- Cookies obtained are not limited by service
  - App may ask for YouTube and then read your email
  - Android permission prompts are misleading
  - i.e. a YouTube token also gives access to GMail
- Prompt is once per app per token type
- Root or physical access is also token access

HOWTO: Hack Google Apps

1. Retrieve *weblogin:* token for domain admin
2. Access domain control panel
   - www.google.com/a/domain.com
3. Get drunk (with power)

Using the Skeleton Key

- Admin *weblogin:* gives a lot of control:
  - Disable 2-Step Verification / Reset Password
  - Reveal Temporary Passwords
  - Create and Modify Privileges/Roles
  - Create/Control Mailing Lists on Target Domain
  - Generate Domain Reports

What About GMail?

- Personal Google accounts are also at risk:
  - Full access to Google Drive, Calendar, GMail, etc.
  - Ability to add recovery address and change password
  - Account setting manipulation for espionage

Ways to Obtain *weblogin:*

1. Legitimate Android Token Request
2. Direct Account DB Query (root access)
3. Physical Device Access (auto sign-in)
4. Extract DB from Device Memory

PoC App Iterations

1. TubeApp: Retrieve Domain OAuth secret
   - Advertised as a YouTube downloader
   - Does not upload credentials

2. Stock View: Steal *weblogin:* tokens
   - Advertised as a Stock Viewer
   - *weblogin:* token is uploaded via HTTP/HTTPS

Stock Viewer PoC Objectives

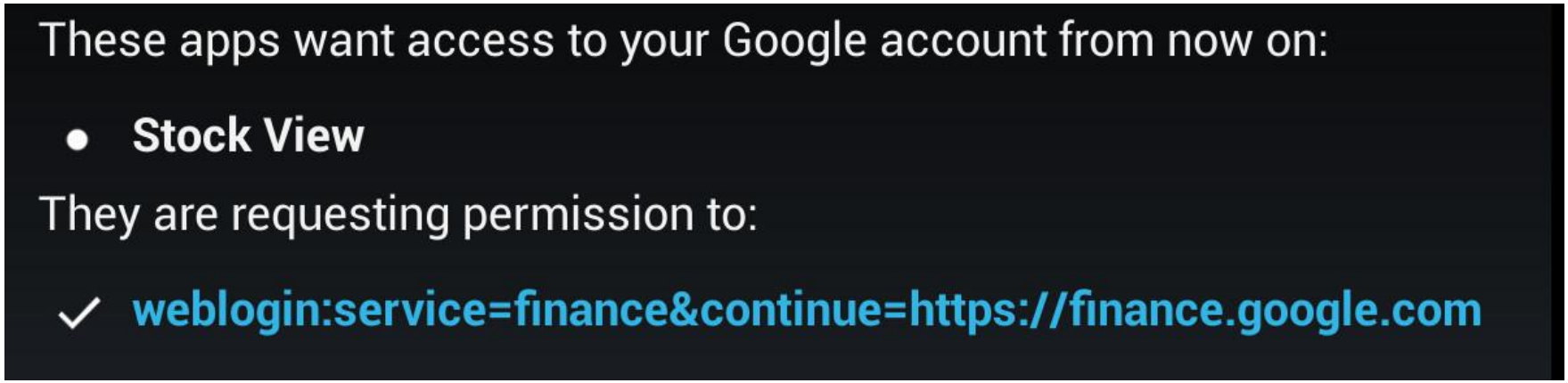
1. Make Token Stealing App without root
   - App requests access to Google Finance (stock ticker)
   - 2 tokens requests == 1 for device + 1 for attacker
2. Publish App in Google Play
   - Will Bouncer allow the token request?
   - Will Bouncer detect that the app is malicious?
3. Scan with Android Security Software
   - Do privacy advisors recognize the threat?
   - Does the token theft get blocked?

Making the App

- Crux of the biscuit:

```
TOKEN_TYPE = \
"weblogin:service=finance&continue=https://finance.google.com/";
getAuthToken(acct, TOKEN_TYPE, null, this, new TokenCallback(), null);
```

- getAuthToken() generates an uninformative prompt:

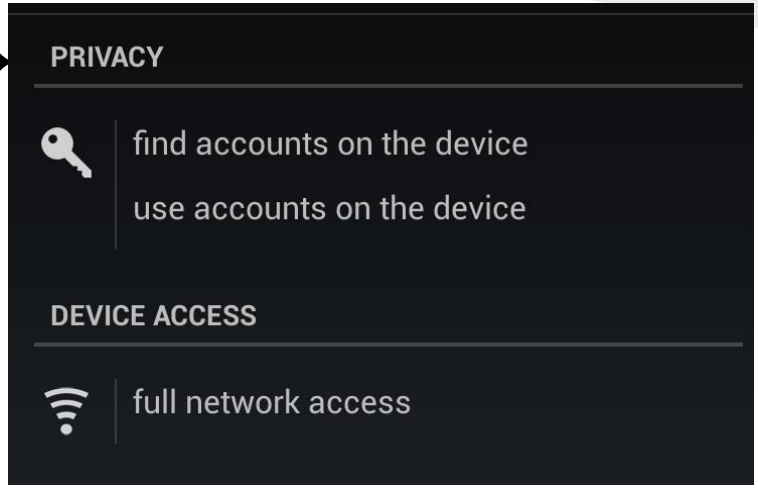


App Revisions

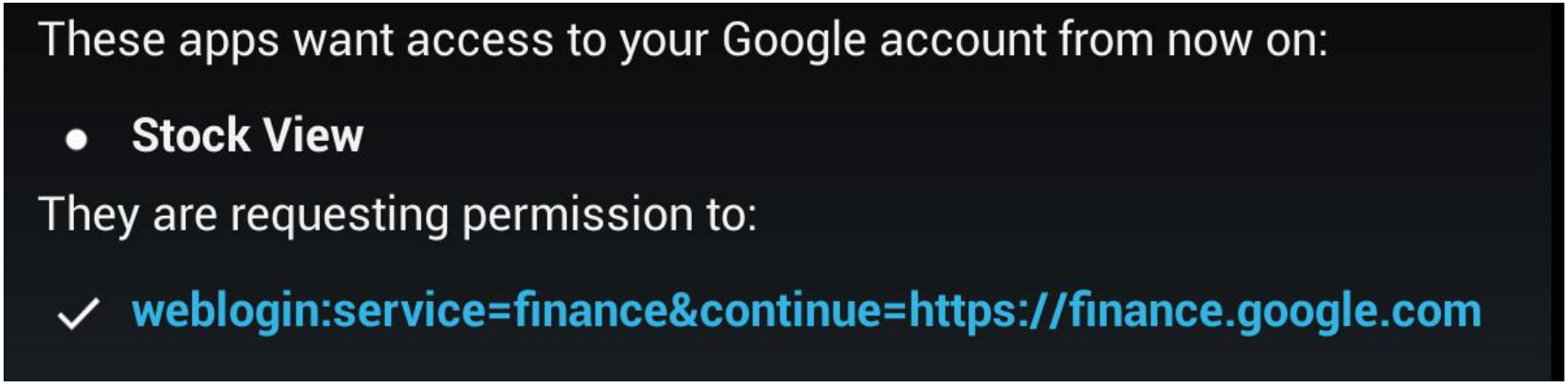
- TubeApp
  - PoC to show OAuth Consumer Secret retrieval
  - Never posted to Play
- Stock View V1
  - Description indicates it is for testing only
  - Price is $150
  - Uploads token if permitted
- Stock View V2
  - Description updated to convey that it is spyware
  - HTTPS added
  - Uploads all available account details
  - Uploads token if permitted

App Permissions

On Install →



On Run ↓



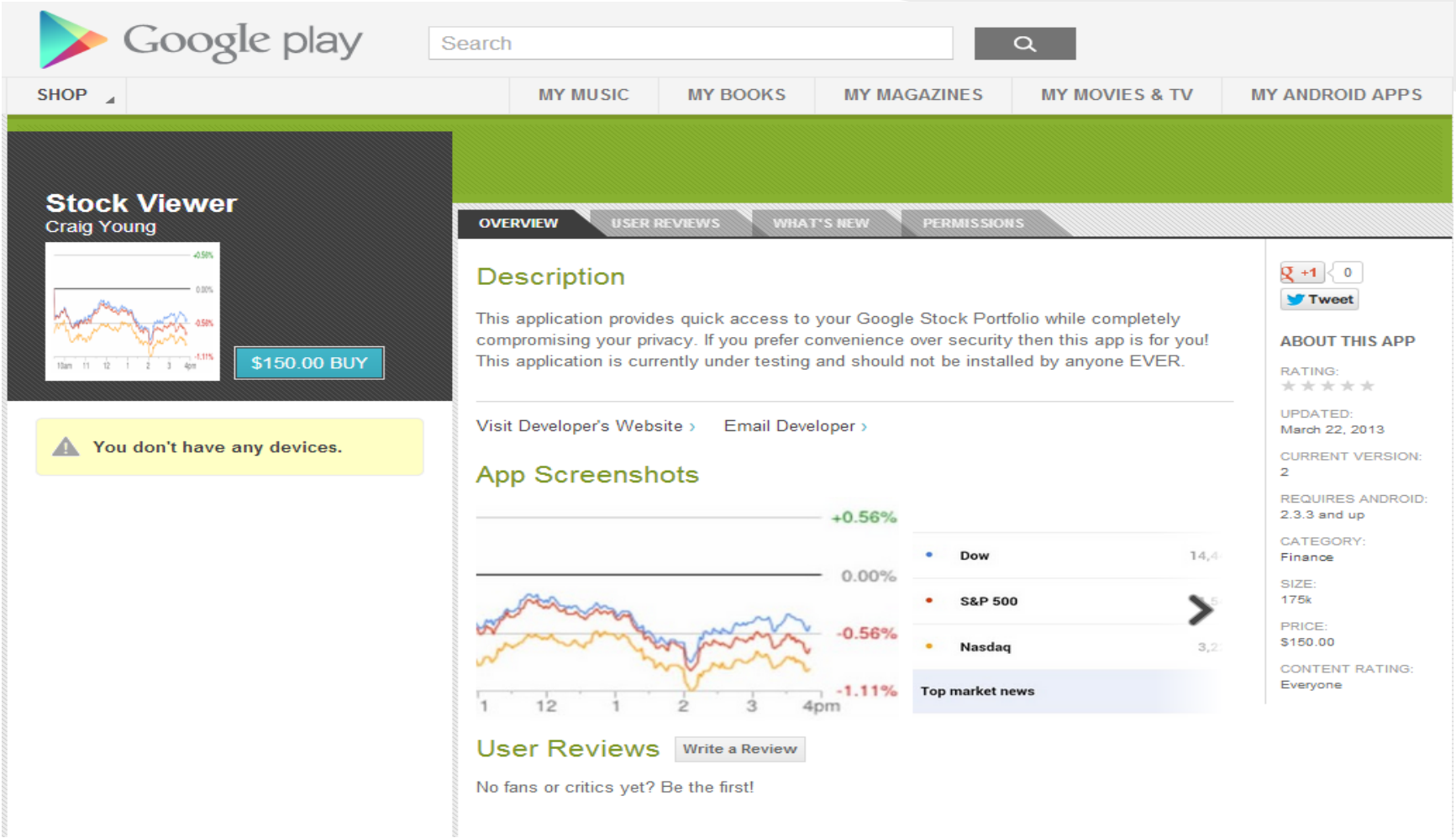
App Results

- Google Play Publication Worked!
  - Nothing was flagged upon submission
  - No data received indicating Bouncer execution

New Questions:

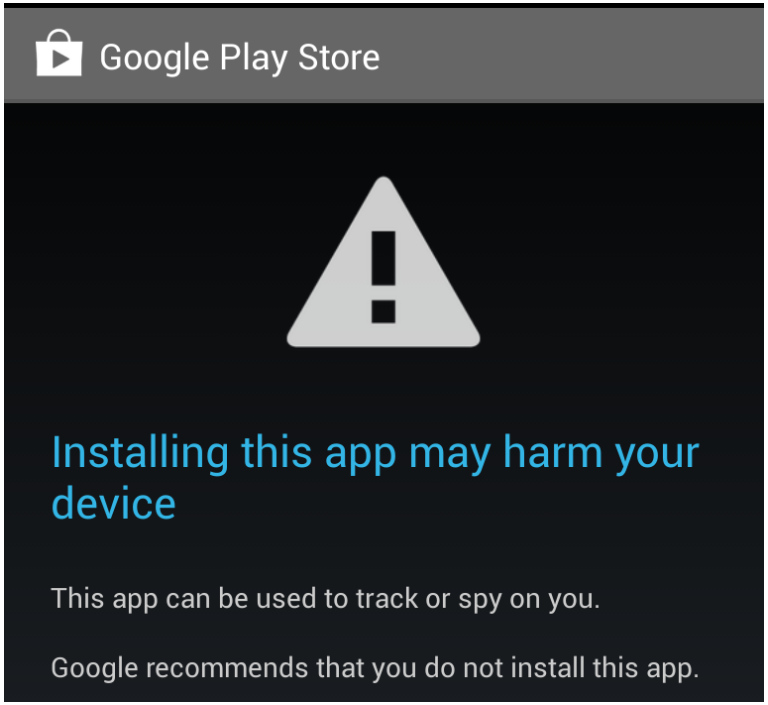
Does Bouncer run all apps?
Does Bouncer run with Google accounts?
Does Google do any manual review at all?

Stock Viewer in Google Play



Play Store Retrospective

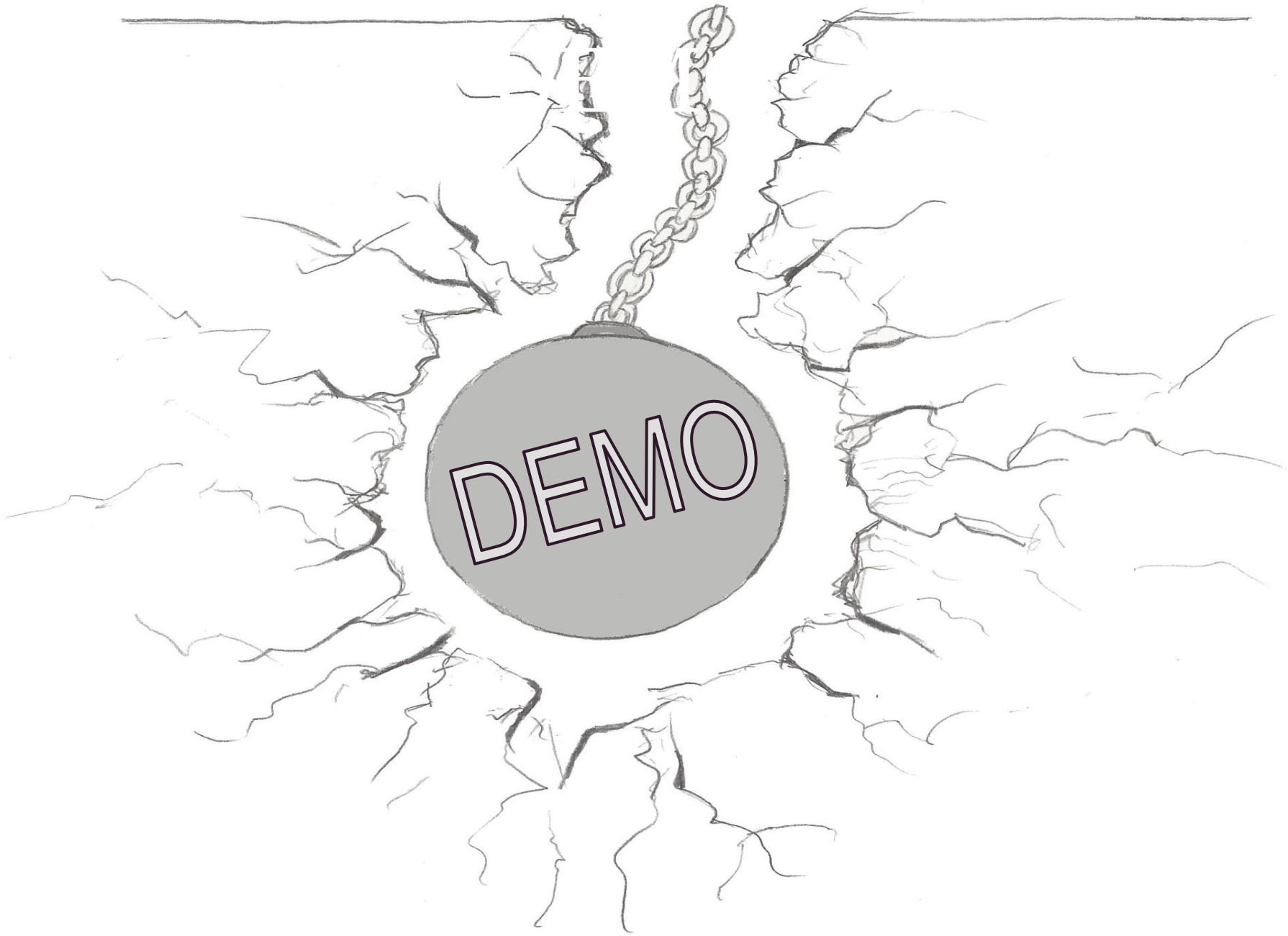
- The app was live on Google Play for a month
- Android Verify now detects it as spyware



End-Point Protection?

- Antivirus/Privacy Advisors
  - Scanned with 5 popular tools
    - Lookout - Safe
    - Norton - No Risk
    - Sophos - Clean
    - Avast - Zero Problems
    - Trend Micro Mobile Security - No Threats Found

- Privacy Advisors
  - Avast Lists it as having account access
  - Lookout Premium did not report access to tokens


DEMO

Don't Be a Victim

- Never use an admin account on Android
- Be very skeptical of token requests
  - *weblogin:* as well as *LSID/SID*
- Avoid downloading apps outside of Play
- Run Antivirus to detect root exploits

Incident Response

- Punt the intruder:
  - Invalidate all sign-in cookies
  - Reset password(s)
- Review affected accounts for:
  - New mail forwarding rules
  - New recovery email address
  - New domain admins
- Analyze Google Apps audit trail:
  - Identify which actions were unauthorized
  - Record IP addresses used by intruder

Further Reading

Here are some helpful references to learn more:

http://nelenkov.blogspot.com/2012/11/sso-using-account-manager.html

https://www.brighttalk.com/webcast/7651/69283

https://blog.duosecurity.com/2013/02/bypassing-googles-two-factor-authentication/

Questions?

Follow @CraigTweets