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 unclear
  - 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

Attacking Google Apps

1. Retrieve *weblogin:* token for domain admin
2. Access domain control panel
3. ???
4. PROFIT!!!!

Using the Skeleton Key

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

*Google tried to fix this for my talk. See demo for details.

What About GMail?

- Personal Google accounts are also at risk:
  - Full access to Google Drive, Calendar, GMail, etc.
  - Ability to reset password (when 2SV is not enabled)*
  - Data dump (Google Takeout)*

* Addressed by Google in response to my talk

More Access

- Remote install of apps from Google Play
- Authenticate through Federated Login
- Create Google Sites

Ways to Obtain *weblogin:*

1. Malware + AccountManager API
2. Query accounts.db (Using root exploit)
3. Physical Access (Chrome auto sign-in)
4. Chip-Off Forensics (Memory extraction)

Stock Viewer PoC Objectives

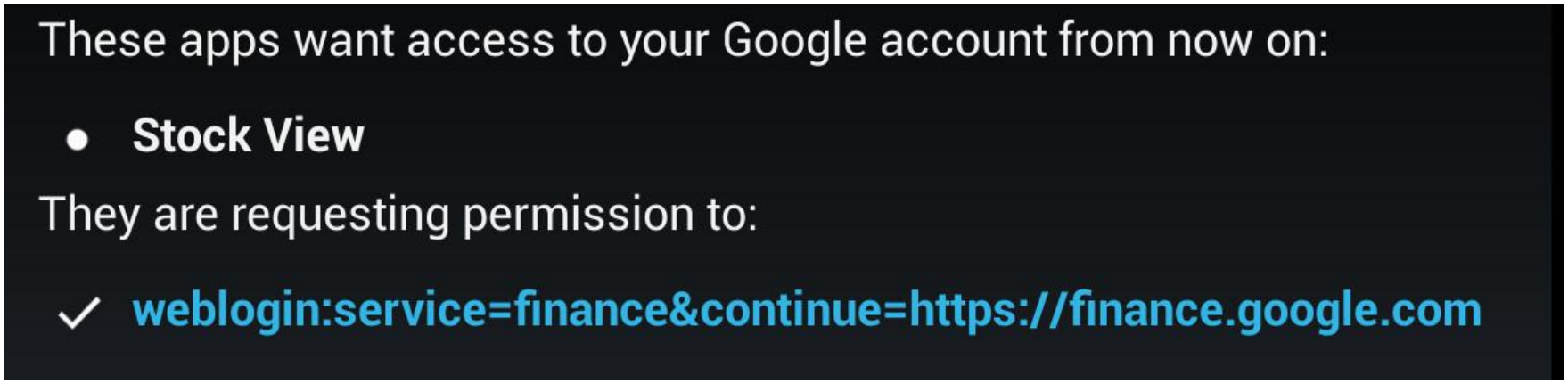
1. Make Token Stealing App without root
   - App requests access to Google Finance (stock ticker)
   - 2 tokens requests == 1 for device + 1 for attacker
2. Publish App in Google Play
   - Will Bouncer allow the token request?
   - Will Bouncer detect that the app is malicious?
3. Scan with Android Security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

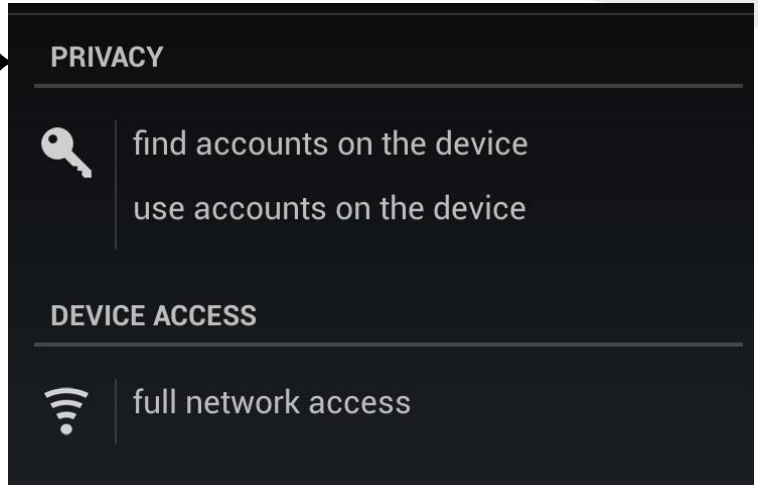


App Revisions

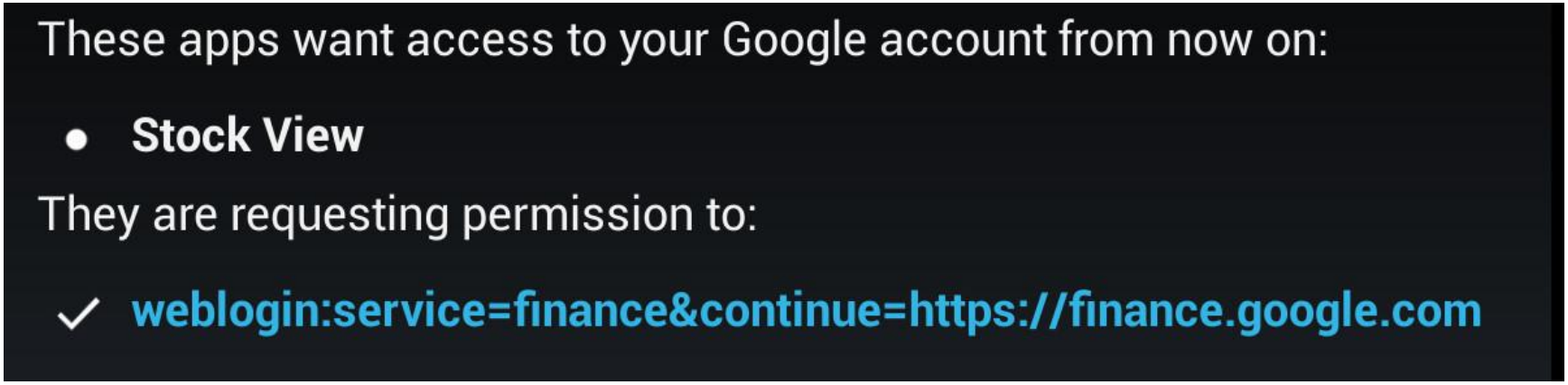
- TubeApp (Quick PoC)
  - Idea is to present as a ‘YouTube’ downloader
  - Fetches domain OAuth secret for in-app display
  - No token egress
- Stock View V1
  - Description indicates it is for testing only
  - Price is $150
  - Uploads token if permitted
- Stock View V2 (SSL Release)
  - Description updated to convey that it is spyware
  - Uploads all available account details
  - Uploads token if permitted

App Permissions

On Install →



On Run ↓



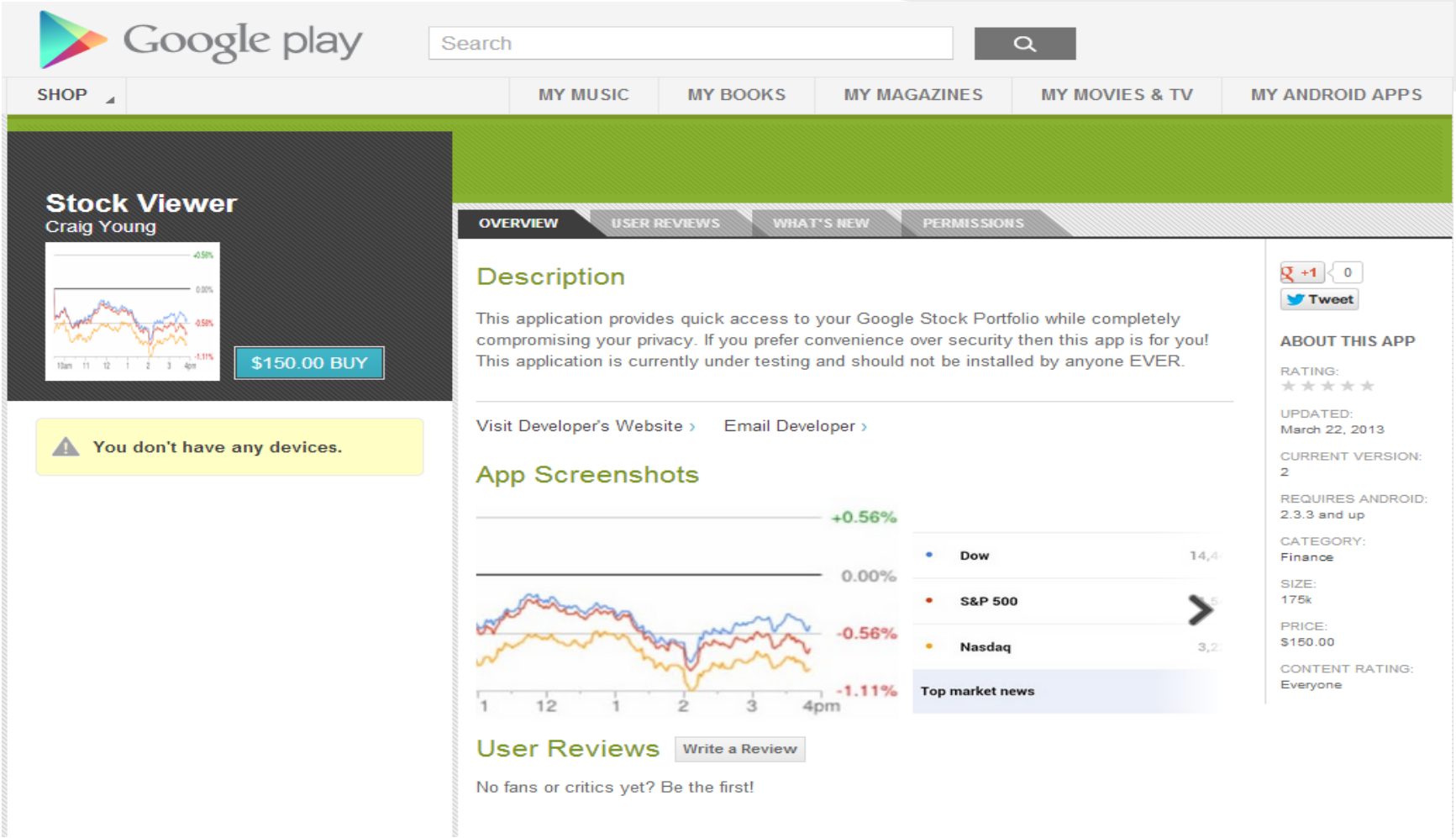
App Results

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

New Questions:

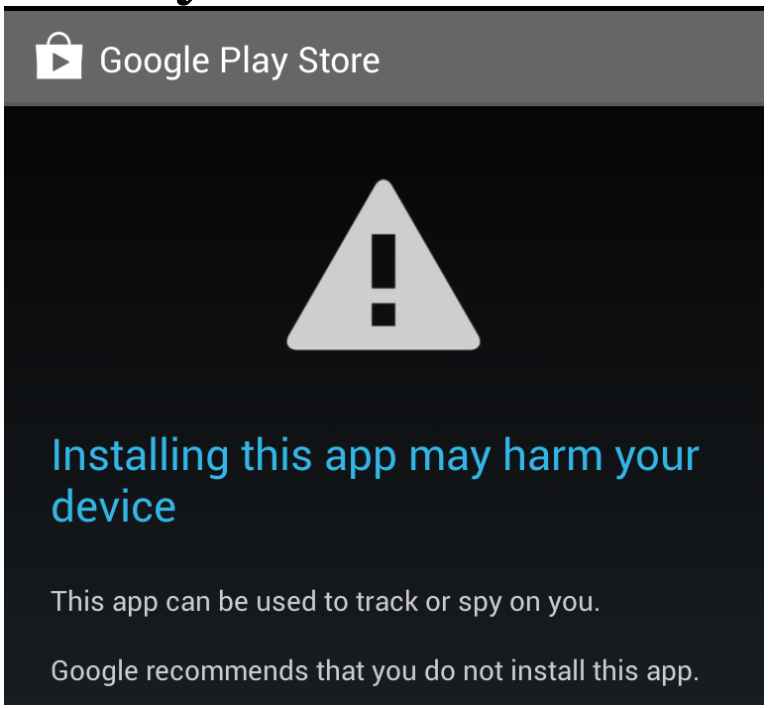
Does Bouncer run all apps?
Does Bouncer run with Google accounts?
Does Google do any manual review?

Stock Viewer in Google Play



Play Store Retrospective

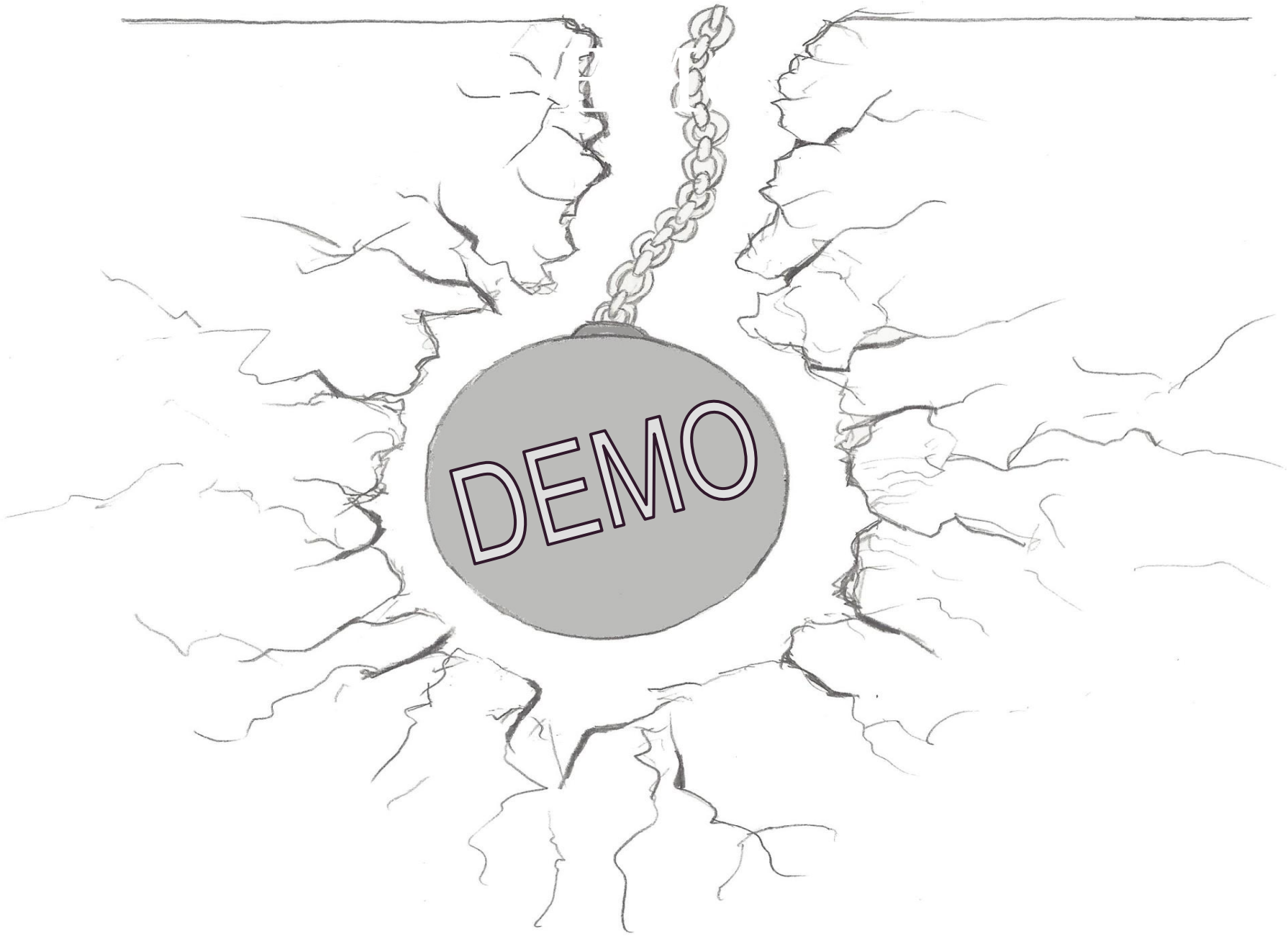
- The app was live on Google Play for a month
- Android Verify now detects it as spyware*



* Unless the app is renamed!

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*
- Stick with ‘trusted’ app stores and vendors
- 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:

Excellent blog on AccountManager: http://nelenkov.blogspot.com/2012/11/sso-using-account-manager.html

My BSides SF 2013 talk on bypassing 2-step verification: https://www.brighttalk.com/webcast/7651/69283

Duo Security blog on bypassing 2-step verification: https://blog.duosecurity.com/2013/02/bypassing-googles-two-factor-authentication/

Questions?

Follow @CraigTweets