# Exploring the Insecurity of Google Account Registration Protocol via Model Checking

Tian Xie, Sihan Wang, Guan-Hua Tu, Chi-Yu Li and Xinyu Lei

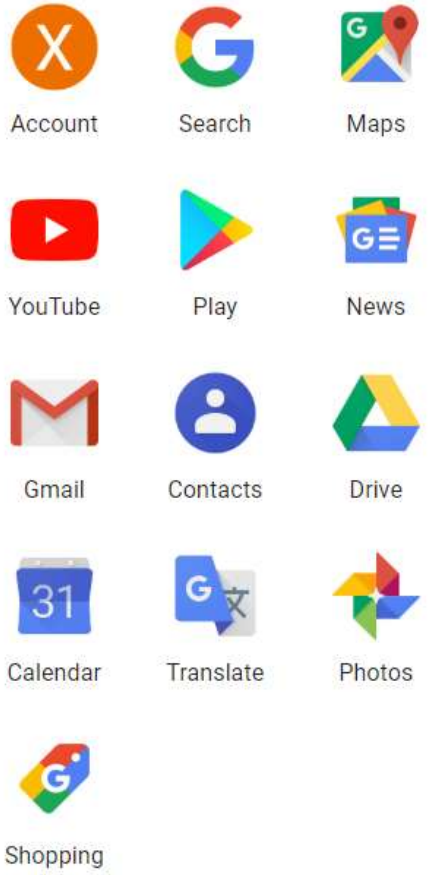
IEEE Symposium Series on Computational Intelligence (SSCI) 2019

# Presentation Outline

- Background
- Google Account Registration Security Mechanism
- Model Checking
- Vulnerabilities
- Attacks
- Solutions
- Conclusions

# Google Account

- Google accounts are essential for many Google services including Gmail, Google Voice, Google Drive, Google Play, etc.



# Dark Side of Google Account

- What people can do with Google accounts?
  - Promote malicious and profitable Android applications using fake reviews and downloads
  - Register faked email accounts
  - Distribute phishing and junk emails

- Google accounts in the dark markets
  - U.S. phone verified Google account: ~$4.5
  - Non-U.S. phone verified Google account: ~$0.4
  - US phone-verified accounts are eight times more expensive than the non-US ones!

# Google Account Registration Security Mechanism

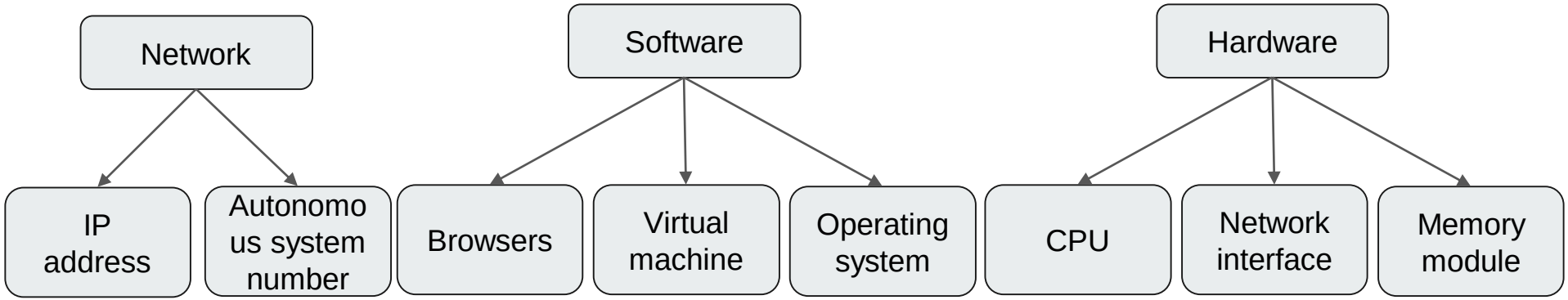
1. Google prevents a device from being used to register 10 or more accounts by identifying the device's fingerprints.
    a. Figerprints are generted by JavaScript codes in the registeration pages.
    b. Google obfuscates the codes to hinder adversaries from forging valid fingerprints.

| |
|---|
| **flowName**: SignUp |
| **gmscoreversion**: undefined |
| **deviceinfo**: [..., "77185425430.apps.googleusercontent.com", "6b8448e8-0c09-45a7-9f49-0671b380290a", ...] |
| **bgRequest**: "web-glif-signup", "!LS6lLg9C9oAl3B9um***qDH5yt04pkJ7q4snnQrqYuxScipeFQ" |
| **f.req**: "AEThLlxg***03iwHkw", "Tyler", "Alvin", "Tyler", "Alvin", "TylerAlvin9426", "11pGVXZ14eEmdu", "TylerAlvin9426", true |
| **continue**: https://www.google.com/ |
| **hl**: en |
| **azt**: AFoagUUW4npIFKevkE-tIb-iNMY4uHJElg:1537326733528 |
| **flowentry**: GlifWebSignIn |

Fig. 2. A decrypted message that carries encrypted device fingerprints (`deviceinfo`).

# Google Account Registration Security Mechanism

1. c. Any change of one of network, software, or hardware settings can’t pass the detection of fingerprints unless all of them are changed.



# Google Account Registration Security Mechanism

2. Google restricts the phone numbers as well as the times for each phone number to verify Google accounts.
    a. Restricted telephone service providers
    b. Restricted phone number reusability

TABLE II
NOT ALL PHONE NUMBERS ARE ELIGIBLE FOR GOOGLE'S PHONE NUMBER VERIFICATION; IT VARIES WITH TELEPHONY SERVICE PROVIDERS.

| Type | Providers | Cost | Supported |
|---|---|---|---|
| Online telephony service providers | Nexmo | $0.57 cents/msg | × |
| | Google Voice | Free | × |
| Mobile network operators | AT&T | $30/month | √ |
| | Verizon | $30/month | √ |
| | T-Mobile | $10 cents/msg | √ |
| | US Mobile | $3.75 cents/msg | √ |

# Google Account Registration Security Mechanism

3. Bot detection
    - When a click event is triggered, the JavaScript codes determine whether it indeed comes from the user by checking whether the cursor's coordinate is located within the button's area on the page by obtaining the mouse cursor's coordinate and calculating the cursor's relative position on the screen.

Are these security mechanisms sufficient to secure the Google account registration?

# GAcctAnalyzer: Google Account Analyzer

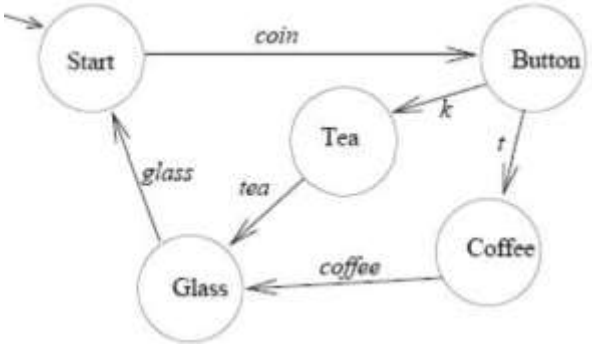
- A **model checking tool** written in Promela (Process Meta Language) to systematically examine the registration security and uncover possible vulnerabilities of the account registration protocol between the clients and Google servers.

# Model checking

- Given a finite-state model of a system
- exhaustively and automatically checking whether this model meets a given specification

Is there such a path? (Start->Button->Tea->Glass->Start)

We can also get exact examples for given specifications



# Model checking

- Advantages
  - Fast detection
  - Support model checking even with partial specifications
  - Support concurrent checking
  - Enumerate counterexamples
- Disadvantages
  - The search space may be too large (e.g., too many paths)
  - Heavy search load when system is too complex

# SPIN model checker

- Advantages:
  - Free, well-documented, actively maintained, large user-base
  - Target software verification
  - The only model checker that  have won the ACM Software System Award
- Features:
  - Use Promela as input language, easy to identify deadlock, infinite loophole, undefined data transfer
  - Use Bit tate Hashing to avoid search entire state space
  - Support for parallel systems and message communication between processes

# GAcctAnalyzer Overview

- Phase 1: Service Screening
- Phase 2: Experimental Validation

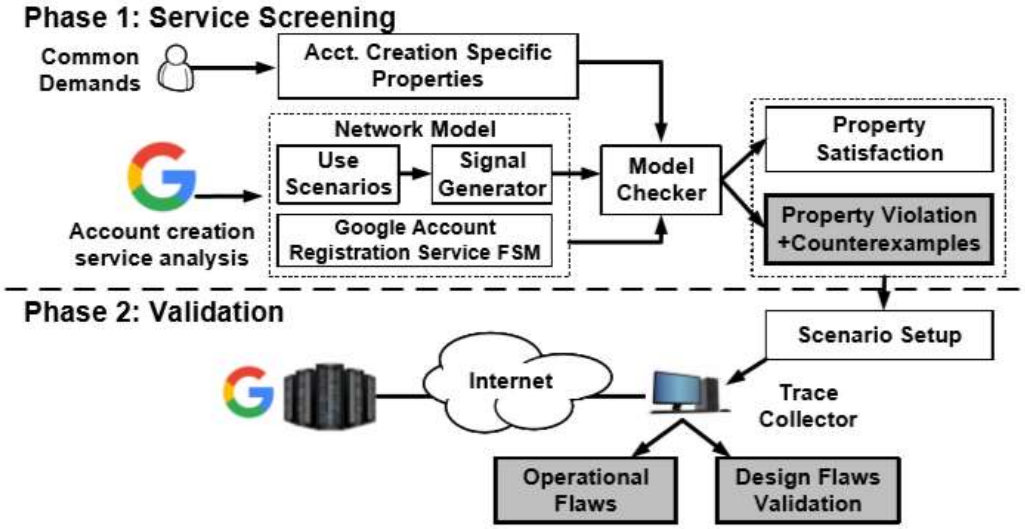
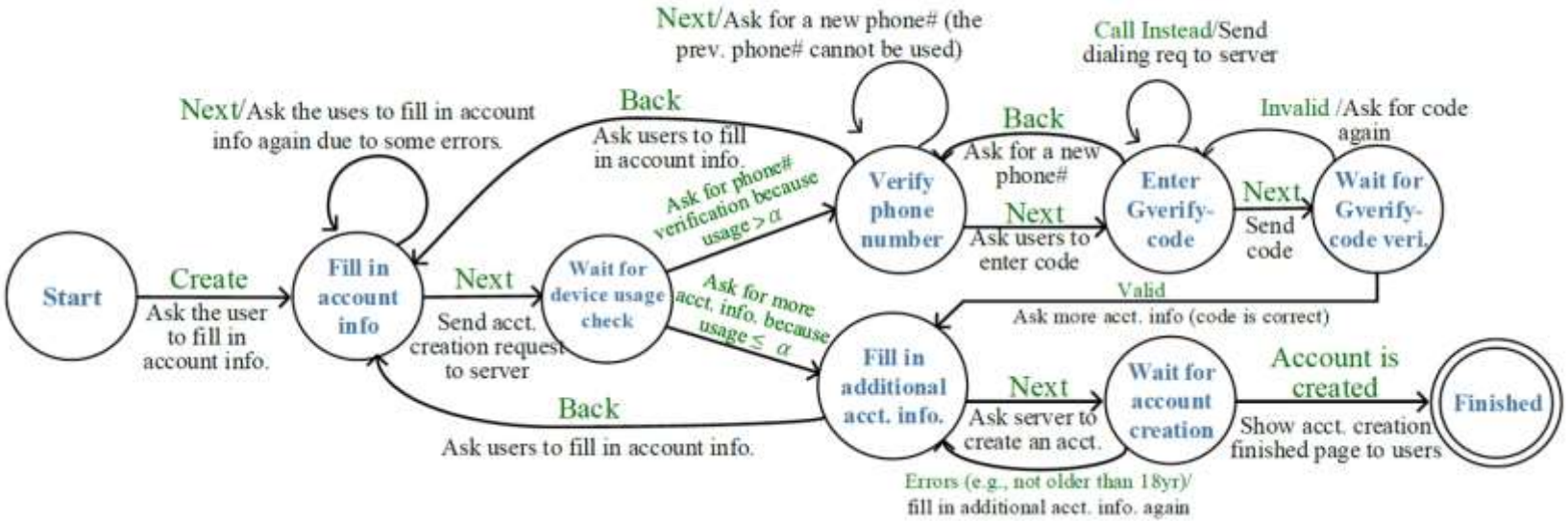
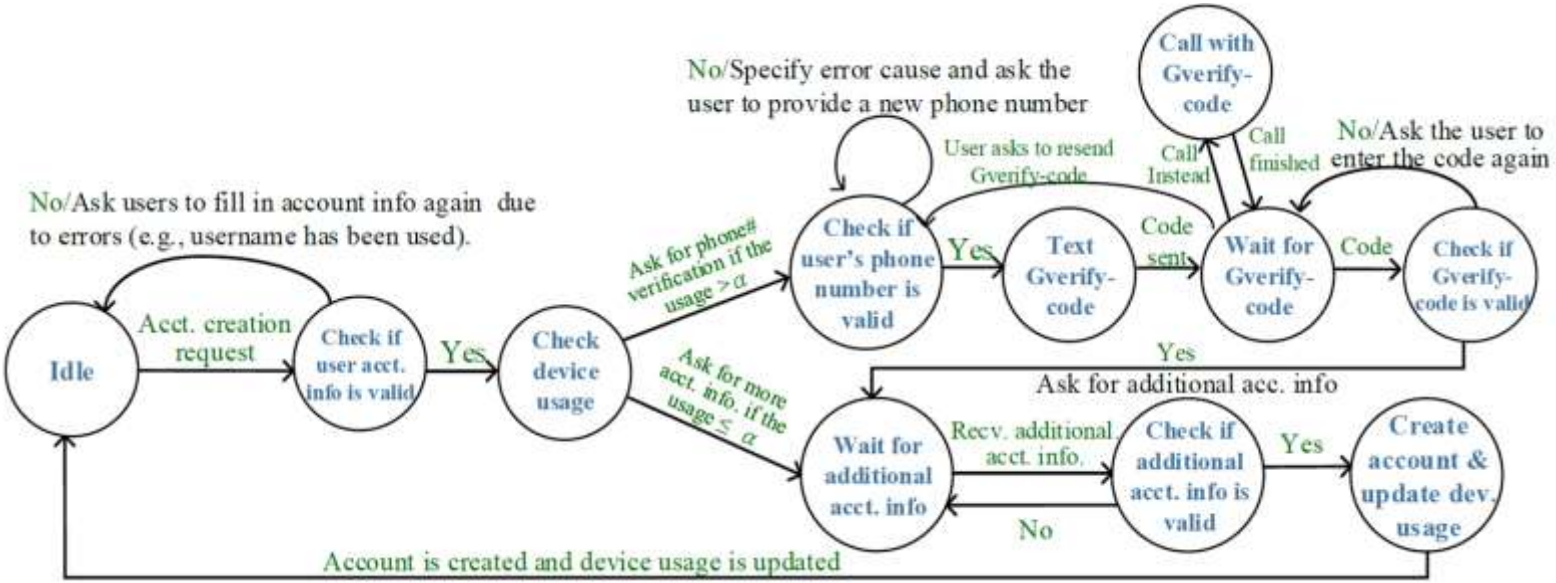



Fig. 3. *GAcctAnalyzer* Overview

- **Client-side** Google account registration



- **Server-side** Google account registration



# Model Checking

- User events: Create, Next, Cancel, Back, Resend, Call Instead, Waiting
- Defined properties:
  - LimitedAccountCreation_WithoutPhoneVerify <=3 per day for one device;
  - RestrictedPhoneNumberUsage <=2  per day for one device;
- Identify property violations
  - SPIN model checker
  - Once a property violation is hit, a counterexample is generated.
- Experimental Validation
  - Try to reproduce counterexamples on the real website.

# Vulnerability1: One-time Check of Device Usage Limit

- Type: Design defect
- Description: The device usage limit (e.g., only 3 accounts without phone number verification can be registered at one device) is checked only once at an intermediate state during the registration process, so a registration instance that passes the check may make the usage exceed the limit after it completes.

# Vulnerability2:  No or Long Inactivity Timeout

- Type: Design defect
- Description: No or long inactivity timeout allows adversaries to make registration instances stay at a certain intermediate state during the registration process and then manipulate them to launch attacks with sufficient time.

# Vulnerability3:  Loose Limits of Phone Number Verifications

- Type: Design defect
- Description: Google places several limits on phone number verifications, but some limits are too loose and may be abused.

- For example, the verification times for a phone number is 10 during 24-hour period.

# Vulnerability4:  Local-view Blockage of Phone Numbers

- Type:  Operation slip
- Description: Google has only local-view blockage of phone numbers for each individual service. Even if one phone number is blocked by the Google account registration service, it can still be successfully used by other Google services.

# Two Proof-of-concept Attacks

- Fake Google Account Generation
- Google Text/Voice Spamming Attack

# Attack1: Fake Google Account Generation

- We devise this attack in two steps.
    - Start a group of registration instances on one device and let them stop at the *Fill_in_additional_acct_info* state. Due to V1, all these instances can pass the one-time check of the device usage limit.
    - Second, make instances proceed to finish their registration procedures. Though it may take a little longer time to fill the required information fields at each registration instance, the registration process has no or long inactivity timeout (V2).
- In this way, each instance can successfully create a Google account without any phone number verification.

# Attack1: Fake Google Account Generation

- Attack variant: low-cost phone-verified accounts (PVAs). We devise this attack variant by leveraging the above attack and V4.

1) We use one number to create 10 PVAs (see V3) and then the number is permanently blocked by Google for the account registration service.

2) We can register another 10 VPAs on another google service (e.g., Google voice).

Reason: Due to V4, although the number is blocked for the account registration service, it is still clean for other google services.

# Attack2: Google Text/Voice Spamming Attack

- During registration process, the attacker can fill in a victim's phone number, so that some spamming emails will send to the victim.
- We develop an attack tool to send text/voice spams as many as possible to victims. The tool keeps a list of victims' phone numbers and generates the allowable spams to them every day. Note that the attack device's IP address will be changed after every 10 spams messages/calls because Google temporarily blocks a device's IP after 10 verification messages.
- According to our experiments, this mechanism does not have the aforementioned anti-spoofing device fingerprinting. Thus, this mechanism can be bypassed by only changing the device's IP address.

# Attack2: Google Text/Voice Spamming Attack

- Evaluation: we use 13 phone numbers from our lab members in the attack test. The numbers are from three US major carriers including Verizon, AT&T, and T-Mobile; the residence of participants covers from the East to the West of the U.S. This attack lasts for one week. Our result shows that each tester indeed receives 70 text messages and 70 voice calls from Google. It confirms that a large-scale attack is feasible since this attack is not limited by carriers or victims' locations.

# Solutions

- Atomic registration process:
    - Google registration process should be limited to an atomic transaction where the check of device usage limit is done right before the completion of the atomic transaction.
    - Due to the atomic transaction, the server can process the request with all users' information and then do the check of device usage limit.

# Solutions

- Anti-spam verification:
    - First, it should reduce the verification limit of a phone number from 10 to a smaller one (e.g., 2 or 3) per day, thereby alleviating the impact of V3.
    - Second, it should provide a way for victims to report the verification spams. For example, the verification text and voice can contain a message that ‘G-XXXXX is your Google verification code. If you did not request it, please reply SPAM.’ Google can thus stop an ongoing attack right away.

# Solutions

- Unified number blockage system:
  - Google should block phone numbers globally with a unified number blockage system.
  - Once a phone number is blocked at one service, this blockage should be propagated to the other services.
  - Google can use a database to maintain the information of blocked phone numbers, and share it with all the services.


Q & A