A Logic-based Security Analyser for Interconnected Online Accounts

Motivations

- With reliance on information technology growing, more and more services are gradually digitised and we use online accounts to handle everything from banking to day-to-day communication.

- However, common information shared across the different online accounts act as links between them, and the compromise of one account can have widespread repercussions.

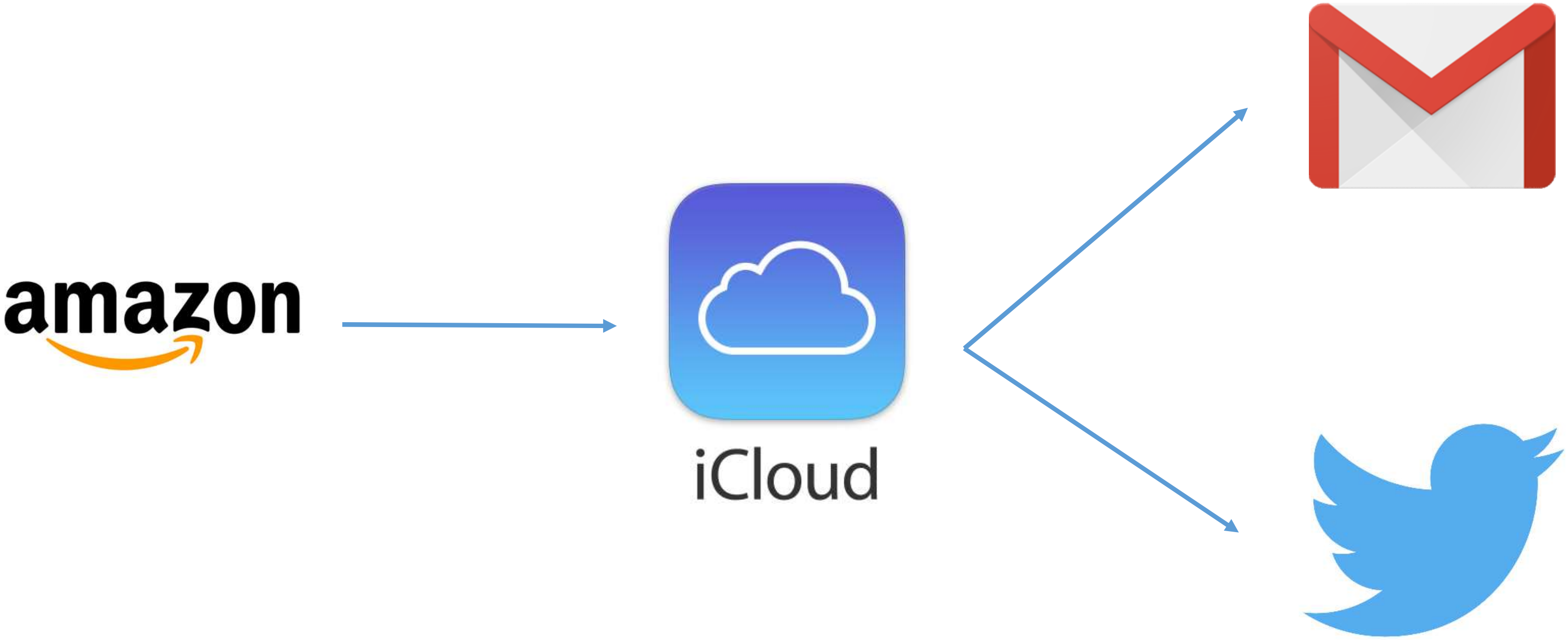
Motivations

- Security questions such as
  - Who is your favourite actor, musician, or artist?
  - What is your favourite book?
  - In what city were you born?
  - What is your date of birth?

- With the popularity of social media platforms such as Facebook, such information is perfectly obtainable.

Motivations

- Matt Honan



Objectives

- Model online accounts, their connections as well as attacks.
- Build preliminary rules for known attacks.
- Identify potential security threats.
- Provide countermeasures to identified threats.

Background

- NuSMV implementation for identifying security flaws in interconnected accounts (Harling)
- Similar works on network security

NuSMV Tool

- Each account expressed as a module with variables such as username and password
- Connectivity expressed by next expressions
- Attack steps are state transitions expressed also using a next expression
- Access policy defined by temporal logic specifications

NuSMV Tool

- Limitations
  - State-explosion
  - Waiting time escalates after running the tool with 25 online accounts (average person has 40
  - Time efficiency

Network Security

- Similarities
    - Hosts = accounts
    - Connectivity
    - Access policy

- Differences
    - Type of exploits

Network Security

- Frameworks
  - Model checking tools
  - MulVAL
  - Attack graphs

MulVAL

- Logic-based framework which models the interaction of software bugs with system and network configurations and analyse security vulnerabilities

- Steps
  - Scan system for configuration
  - Match known vulnerabilities
  - Encode information in Datalog
  - Captures system interactions using a set of pre-defined rules
  - Analysis of system security

Attack Graphs

- Visual representation of how an exploit, or a series of exploits can affect different hosts in a network.
  - How: state-based attack graph
  - Why: logical attack graph

- State-based attack graphs: each path in the attack graph correspond to a step in an exploit leading up to an undesirable state.
- Logic-based attack graphs: every node is a logical statement, not every single state property of a network is encoded in the node.

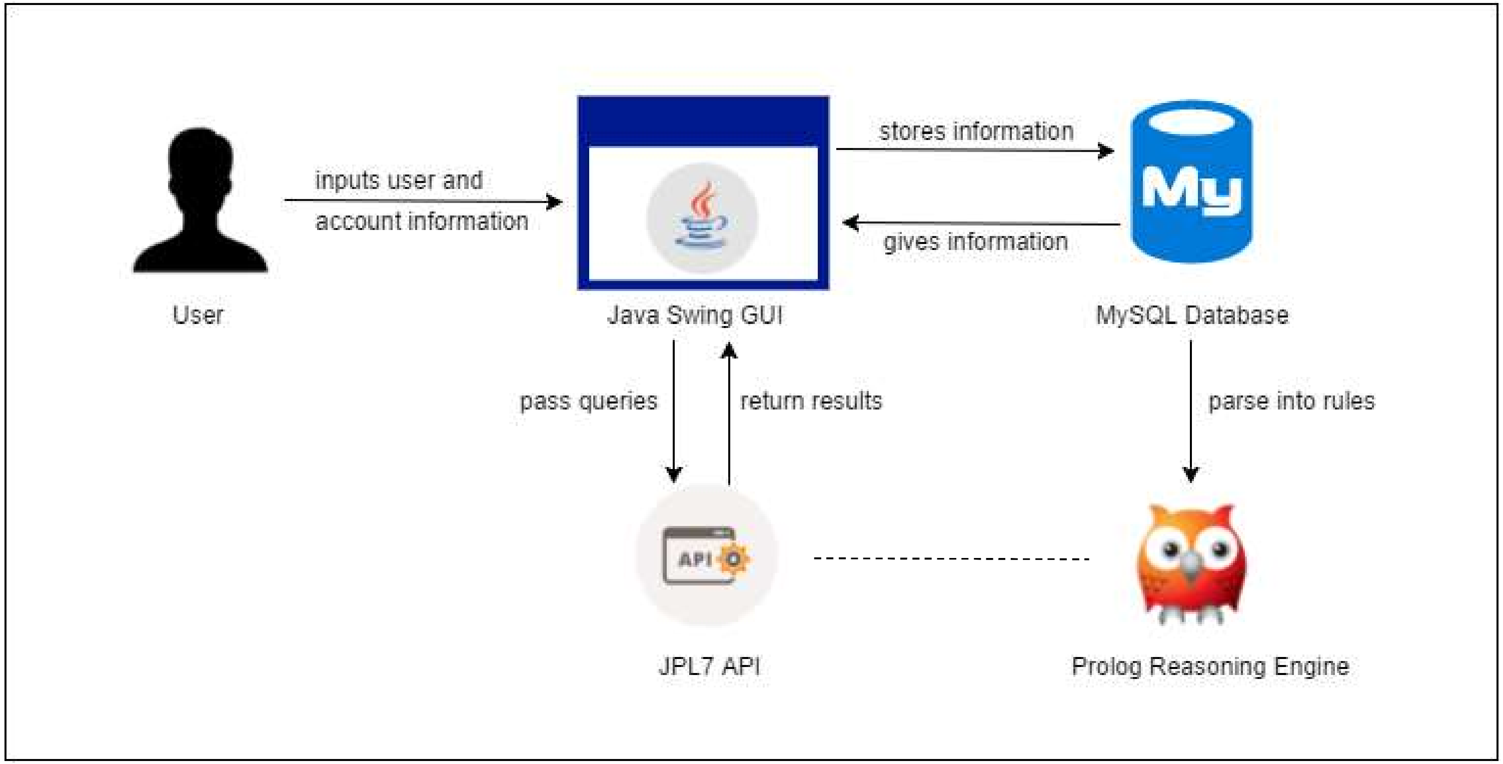
Summary

	Model Checking	**MulVAL**	**Attack Graphs**
+	Support partial verification Diagnostic information	Diagnostic information Efficient	Visual representation Diagnostic information Analysis
-	State explosion Inefficiency		Side-product Time consuming

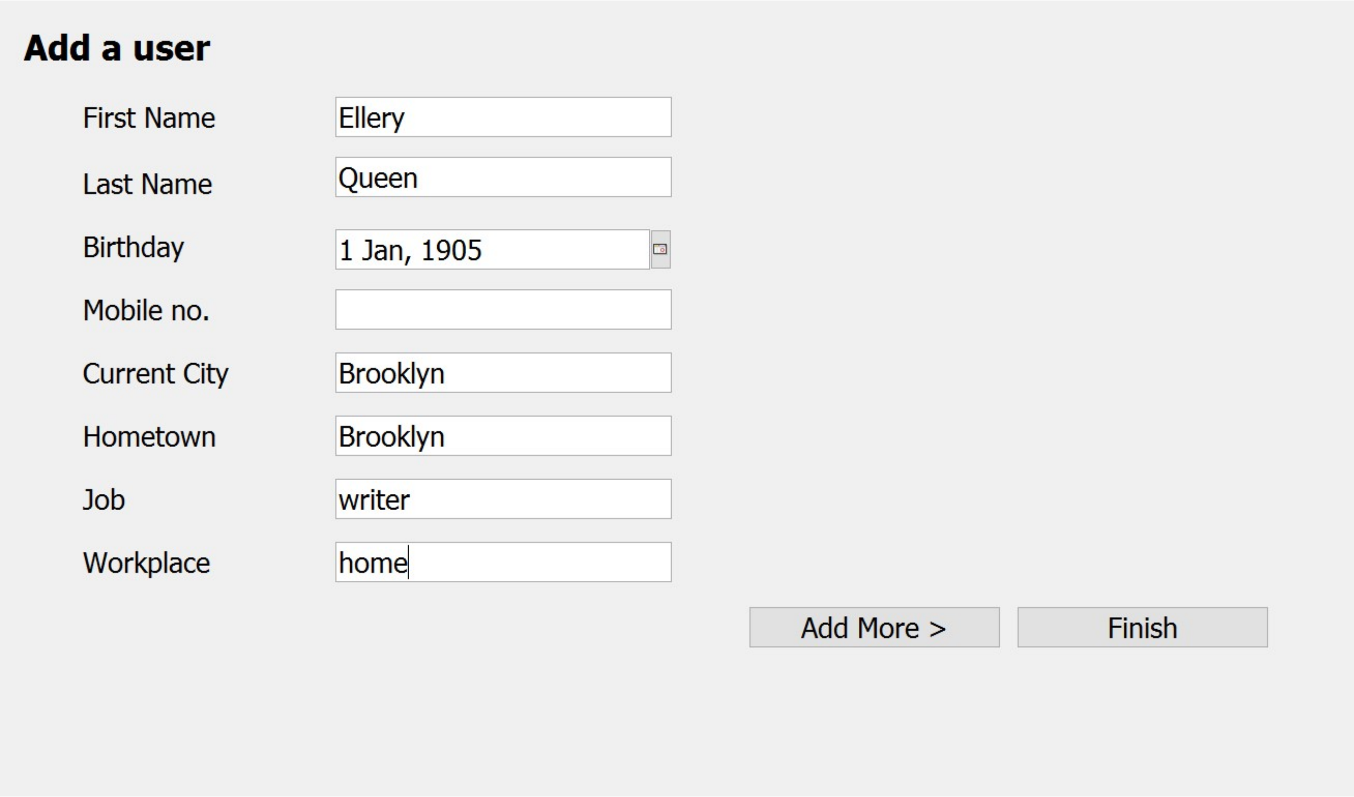
Proposed Solution

- Build tool based upon MulVAL framework
  - For efficiency
- Incorporate attack graphs as an option for the user
  - For in-depth analysis

System Architecture



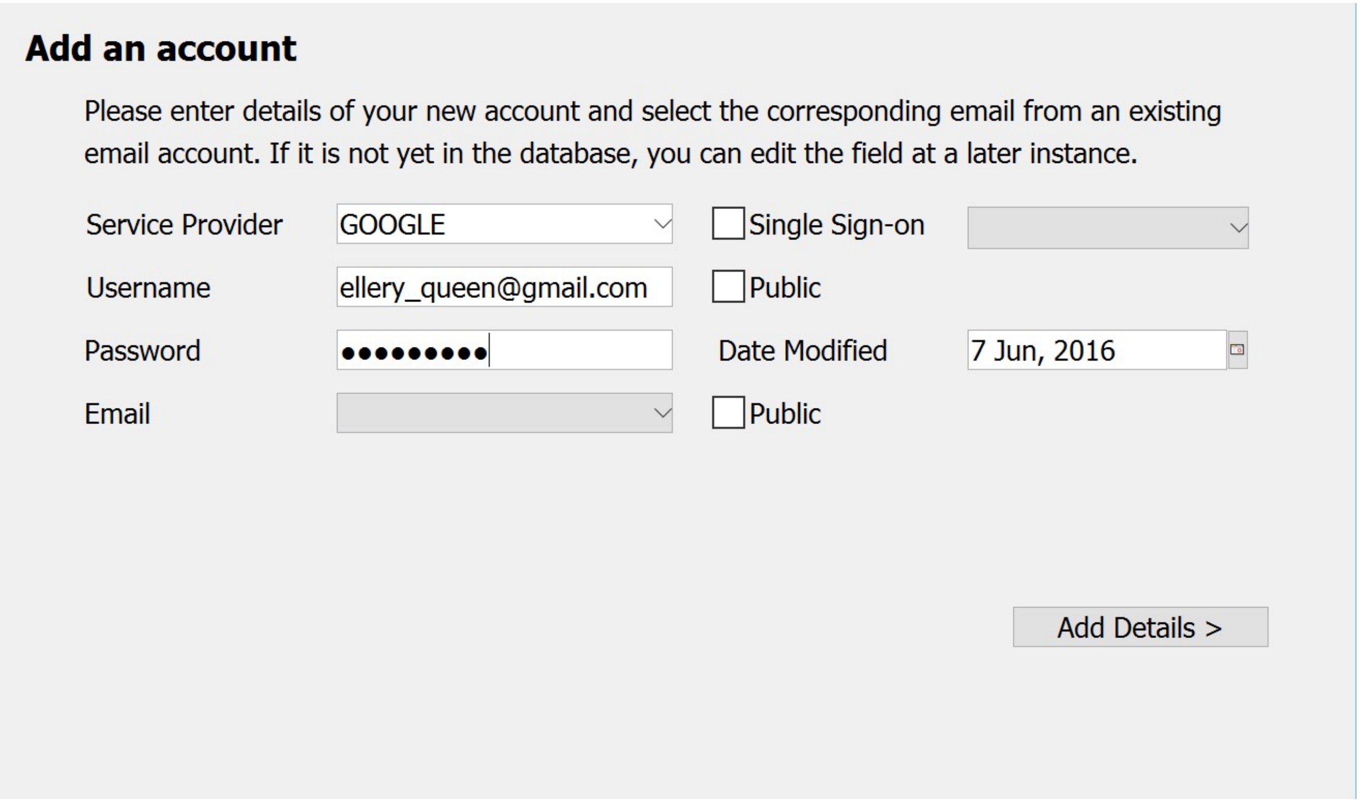
Demo: User Input



Add a user

First Name	Ellery
Last Name	Queen
Birthday	1 Jan, 1905
Mobile no.	
Current City	Brooklyn
Hometown	Brooklyn
Job	writer
Workplace	home

Demo: User Input



Add an account

Please enter details of your new account and select the corresponding email from an existing email account. If it is not yet in the database, you can edit the field at a later instance.

Service Provider: GOOGLE

□ Single Sign-on

Username: ellery_queen@gmail.com

□ Public

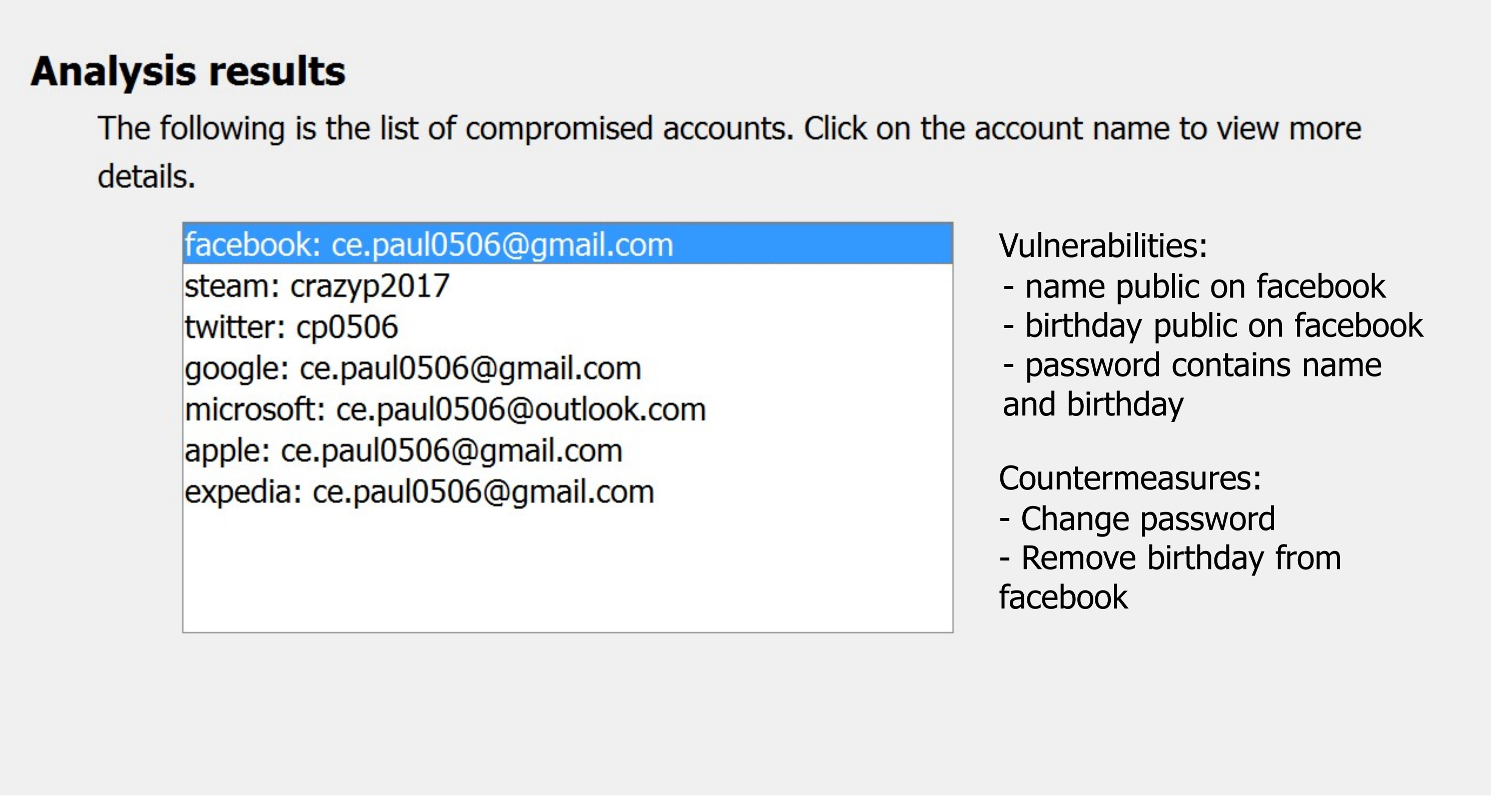
Password: ●●●●●●●●●

Date Modified: 7 Jun, 2016

Email:

□ Public

Demo: Analysis



Analysis results

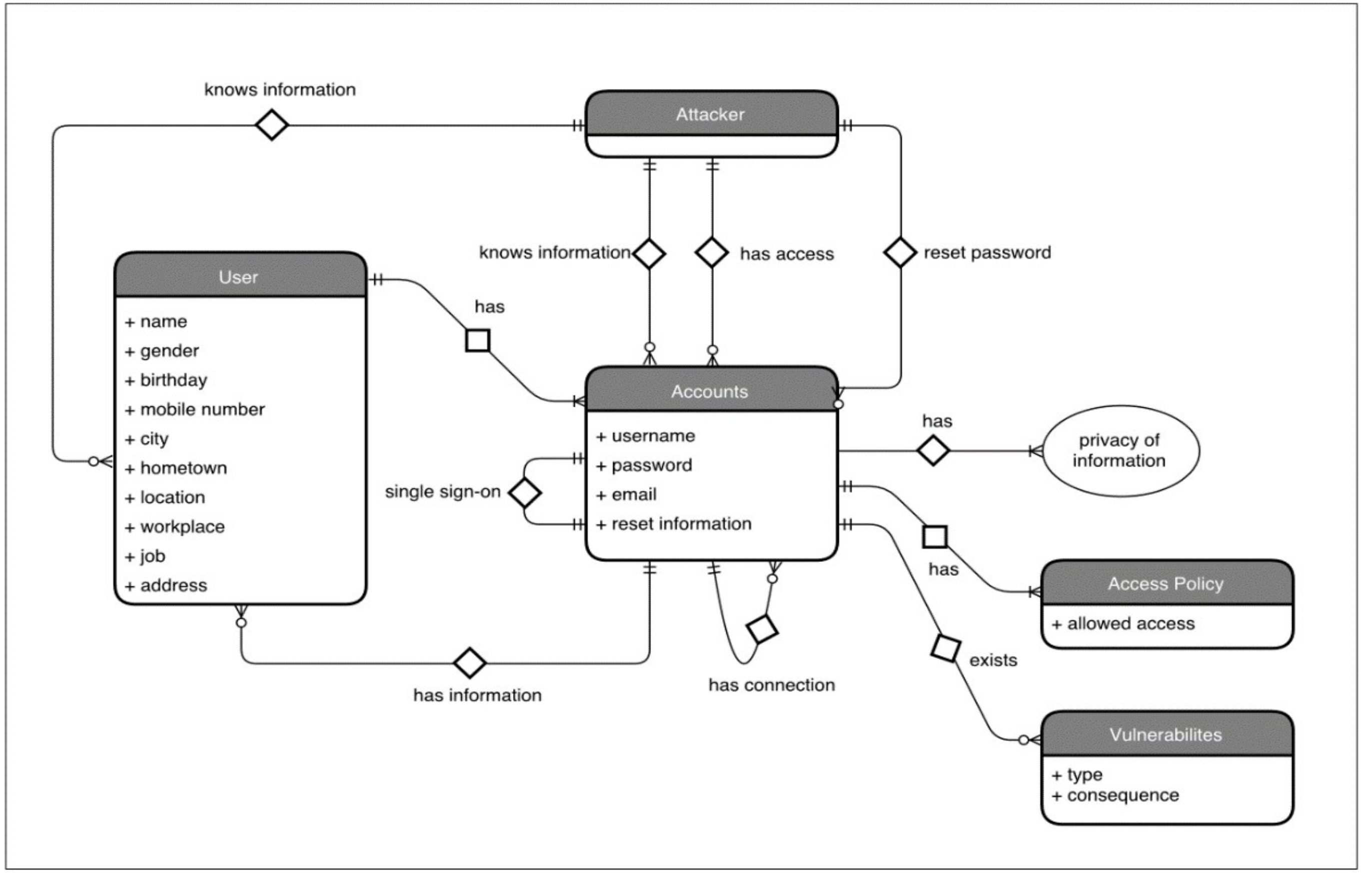
The following is the list of compromised accounts. Click on the account name to view more details.

facebook: ce.paul0506@gmail.com
steam: crazyp2017
twitter: cp0506
google: ce.paul0506@gmail.com
microsoft: ce.paul0506@outlook.com
apple: ce.paul0506@gmail.com
expedia: ce.paul0506@gmail.com

Vulnerabilities:
- name public on facebook
- birthday public on facebook
- password contains name and birthday

Countermeasures:
- Change password
- Remove birthday from facebook

Framework


knows information
Attacker
knows information
has access
reset password
User
+ name
+ gender
+ birthday
+ mobile number
+ city
+ hometown
+ location
+ workplace
+ job
+ address
has
Accounts
+ username
+ password
+ email
+ reset information
has
privacy of information
single sign-on
has
Access Policy
+ allowed access
exists
Vulnerabilites
+ type
+ consequence
has information
has connection

User

- Every user has fixed attributes.
- Other fields can be easily added as well.
- Every account the user owns consists of, to some extent, personal information which identifies the user.

Accounts

- An account can be uniquely identified by its service provider and username.
- Own authentication procedure, or a single sign-on system.
- Every account is also associated with an email address, which is often used for password recovery.
- They also have different password recovery practices.
- Each account contains information about the user classifed as public or private depending on the access level of these information to visitors.

Vulnerabilities

- Vulnerabilities for individual accounts include:
  - Publicly-available username / email
  - Commonly-used passwords
  - Password contains name of user
  - Password same as username of user
  - Weak password
  - Password unchanged for too long

- Vulnerabilities which involves linkage across accounts include:
  - Repeated passwords
  - Repeated usernames
  - Information required for password reset available publicly on another account

Access Policy

- Every account has an access policy which states who should be able to access the account. Anyone outside of this group should not be able to access the account, or it will be considered a violation of this policy.

Attacker

- The attacker carries out exploits using the information known from the account.

- There are three main types of exploits:
  - Compromise an account with username and password
  - Access to the SSO account if account uses SSO verification
  - Reset password

Example Database

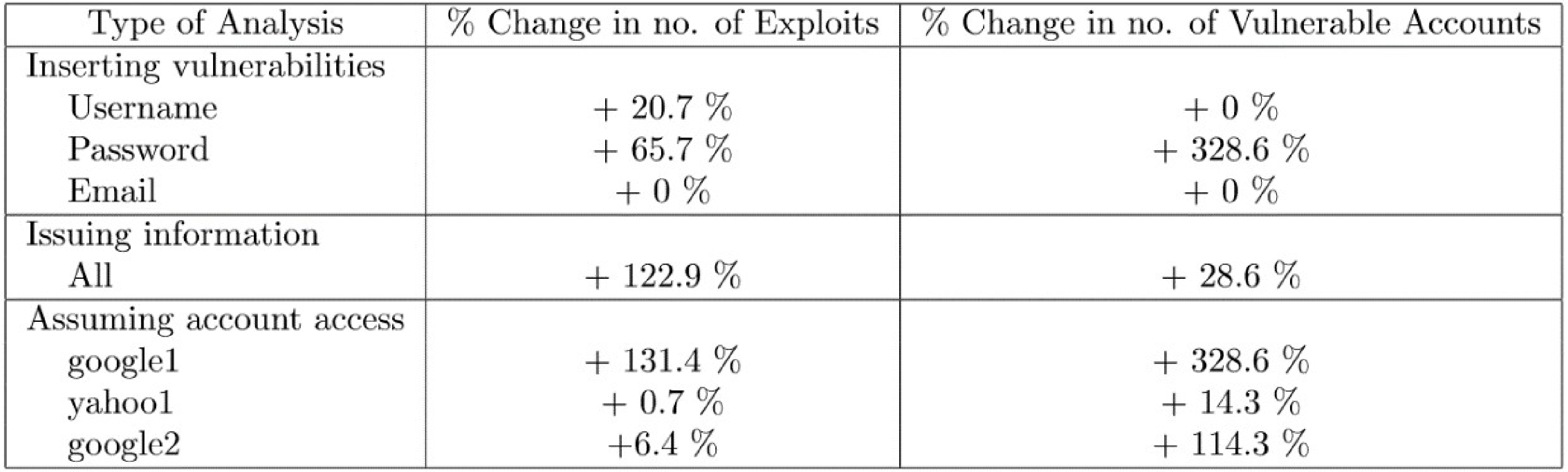
- 35 online accounts, chosen from a selection of the most popular websites in the world today.
- Vulnerabilities are injected into these accounts, and rules for interactions are set up.
- Possible exploit measures are taken into account, and then we check for violations to security policies and generate corresponding countermeasures.

Hypothetical Analysis

- The types of hypothesis analysis we have implemented are:
  - Inserting vulnerabilities
  - Issuing information
  - Allowing account access

Hypothetical Analysis

- Results of hypothetical analysis



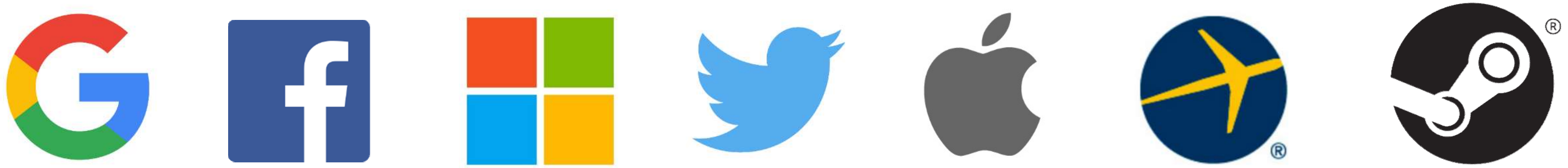
Type of Analysis	% Change in no. of Exploits	% Change in no. of Vulnerable Accounts
Inserting vulnerabilities		
Username	+ 20.7 %	+ 0 %
Password	+ 65.7 %	+ 328.6 %
Email	+ 0 %	+ 0 %
Issuing information		
All	+ 122.9 %	+ 28.6 %
Assuming account access		
google1	+ 131.4 %	+ 328.6 %
yahoo1	+ 0.7 %	+ 14.3 %
google2	+6.4 %	+ 114.3 %

Countermeasures

- Password related vulnerabilities
- Repeated passwords
- Information required for password reset available publicly on another account
- Others

Attack Graphs

- Attack graphs provide a good way of visualising the possible exploits, but the connections become complicated when more accounts are used, thus we use a reduce sample of 7 accounts as an example instead



Attack Graph

Attack Graph Metrics

- Non-path metrics

Metric	**Value**
Network Compromise Percentage	57%
Weakest Adversary	Twitter

Attack Graph Metrics

- Path metrics

Metric	**Value**
Shortest path	4 (Twitter)
No. of paths	14
Mean path length	7.9
Standard deviation	2.43
Mode	10
Median	10
Normalised mean	0.29

Attack Graph Countermeasures

- Minimum Critical Set
  - Remove email address of the user from the public information made available on Facebook
  - Change password of the Twitter account
- Single Action Removal
  - Remove email address of the user from the public information made available on Facebook

Extensions

- Security of information input into tool
- Additional support for greater variety of exploits
- Standalone app included with dependencies

Conclusion