Università degli Studi di Padova

SPRITZ Security & Privacy Research Group

HUMAN INSPIRED TECHNOLOGY Research Centre

DIPARTIMENTO MATEMATICA

ACM SAC ‘19 - Privacy by Design in Practice Track (PDP)

Mind Your Wallet's Privacy

Identifying Bitcoin Wallet Apps and User's Actions through Network Traffic Analysis

Fabio Aiolli, Mauro Conti, **Ankit Gangwal,** Mirko Polato
Department of Mathematics, University of Padua, Italy

Table of contents

Introduction

- Bitcoin [1] offers pseudo-anonymity, which is, unfortunately, being exploited to commit several financial frauds.
  - E.g., ransomware campaigns.

- Our work primarily aims to identify/filter smartphone-based Bitcoin wallet users.

- We use network traffic analysis using machine learning techniques to achieve our goals.
  - *The governments can ask telecom operators to identify/filter Bitcoin wallet users using our approach, which can assist in the hunt of cyber-criminals.*

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” https://bitcoin.org/bitcoin.pdf, 2008.

Introduction

- Our approach works even when the payload is encrypted.

- Using network traffic analysis, we identify:
  - If a user is using a Bitcoin wallet app or not?
    - If so, then which app the user is using?
  - What actions they performing on these apps?
    - Sending, receiving, etc.

- Our work can be extended to other categories of smartphone apps to further improve user profiling.

Smartphone, app, and action selection

- According to Gartner [2], Android and iOS devices together accounted for 99.9% of all smartphone sales by the end of the year 2017. Hence, in our study, we used both Android and iOS devices.

- For the apps, we chose the worldwide most downloaded [3] Bitcoin wallet apps on both Google Play Store and Apple’s App Store in the year 2017.

- For non-Bitcoin apps, we chose the top-10 apps along with additional 20 Internet-dependent apps from each store.

[2] gartner.com/newsroom/id/3859963
[3] sensortower.com/blog/bitcoin-wallet-app-growth

Smartphone, app, and action selection

- We found seven classes of actions relevant to Bitcoin transactions.

- The most important (and common across each app) actions for Bitcoin transactions are:
  a. Send Bitcoin,
  b. Receive Bitcoin,
  c. Open the app (sync data).

App	Action						
	Open app	Receive Bitcoin	Send Bitcoin	Generate addresses	In-app buy/sell	Transaction history	Check balance
BTC.com	✓	✓	✓	✗	✠	❄	❄
BitPay	✓	✓	✓	✓	✓	♣	❄
Bitcoin Wallet (Bitcoin Wallet Devs)	✓	✓	✓	✗	✗	❄	❄
Bitcoin Wallet (Bitcoin.com)	✓	✓	✓	✓	✠	♣	❄
Blockchain	✓	✓	✓	✗	✓	♣	❄
Bread	✓	✓	✓	✗	✓	♣	❄
Coinbase	✓	✓	✓	✗	✓	♣	❄
Copay	✓	✓	✓	✓	✓	♣	❄
Luno	✓	✓	✓	✗	✓	❖	✙
Mycelium	✓	✓	✓	✗	✠	❖	❄
Unocoin	✓	✓	✓	✗	✓	❖	❄
Wirex	✓	✓	✓	✗	✓	❄	❄
Xapo	✓	✓	✓	✗	✓	❖	❄
Zebpay	✓	✓	✓	✗	✓	❖	❄

✓ Available ✗ Not available ❄ On app's home ♣ Under individual wallet/currency
✙ Under dedicated menu for wallets' summary ❖ Under dedicated menu for transaction history
✠ Redirects to an external website, leaving the app

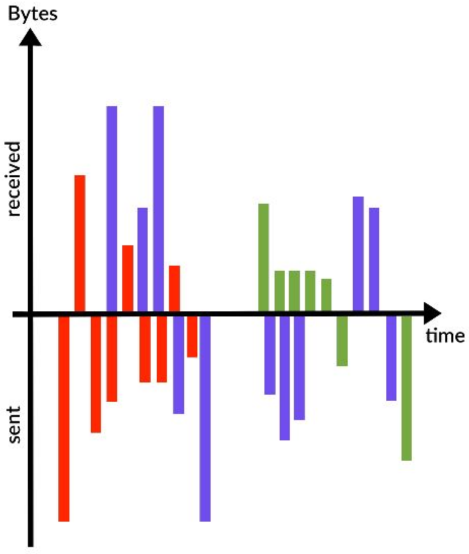
Classifier design

- Our classification procedure consists of the following steps:
  a. Data preprocessing
  b. Feature selection
  c. Machine learning
  d. Training & prediction

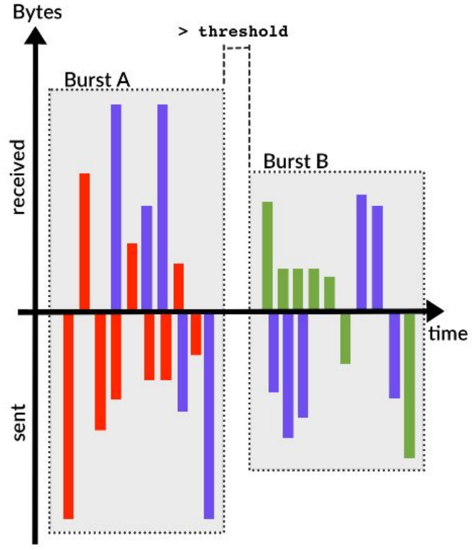
Classifier design

Data preprocessing

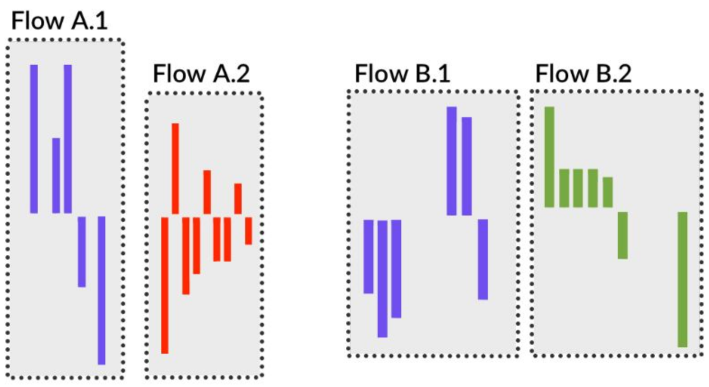
- To handle network traffic traces via machine learning models, we need to perform a preprocessing step, which includes:
  a. Network trace capture
  b. Traffic *burstification* (time/threshold based)
  c. Flows separation



(a) Network traces capture: different colors represent different applications.



(b) Traffic *burstification*: traces are split into bursts.



(c) Flows separation: for each burst, different flows are separated by means of the pairs of source-destination IPs.

Classifier design

Feature selection

- Now, we convert the flows into training instances using following 12 statistics: *length of the series, minimum, maximum, mean, median, mode, variance, skewness, kurtosis, and percentile (25%, 50% and 75%).*

- These statistics are computed for: *the entire sequence, incoming packets only, and for the outgoing packets only.* Hence, the resulting instance has a dimension of 36 (12 × 3).

Classifier design

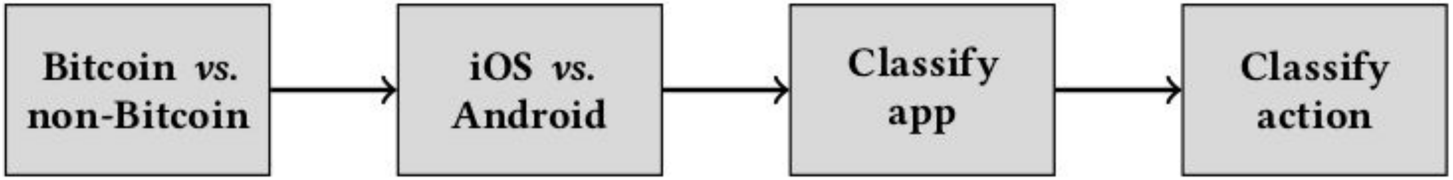
Machine learning

- We employed two most successful machine learning methods for classification:
  a. Random Forest (RF)
  b. Support Vector Machine (SVM)

Classifier design

Training & prediction

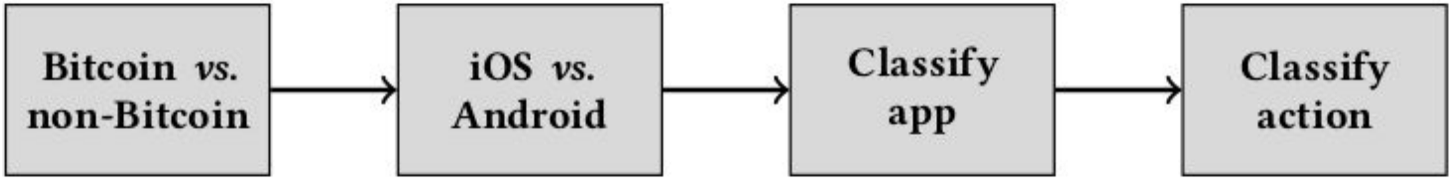
- Training: We tackle our problem at different levels. We identified the following layers of classification:
  a. Classify whether the instance represents a flow of a Bitcoin app;
  b. If so, classify whether it belongs to an Android app or an iOS app;
  c. If it has been categorized as Android/iOS app, classify specific app;
  d. Given the app from the previous step, finally, classify the specific action.



Classifier design

Training & prediction

- Training: We tackle our problem at different levels. We identified the following layers of classification:
  a. Classify whether the instance represents a flow of a Bitcoin app;
  b. If so, classify whether it belongs to an Android app or an iOS app;
  c. If it has been categorized as Android/iOS app, classify specific app;
  d. Given the app from the previous step, finally, classify the specific action.



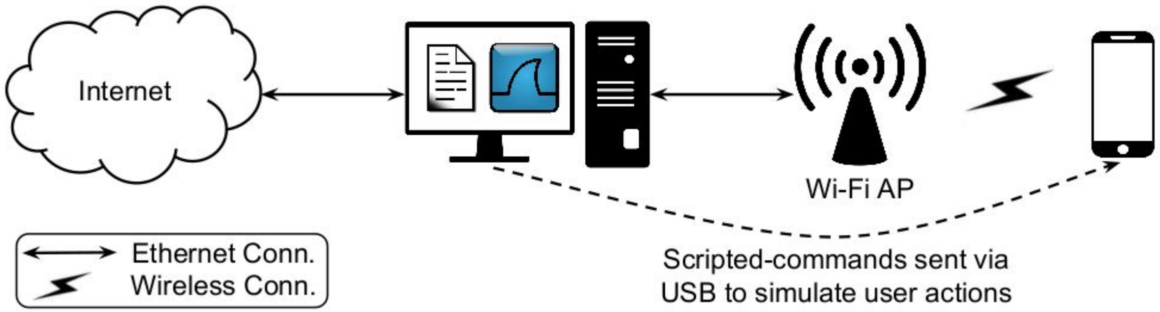
- Prediction: Given a new instance to classify, the prediction is performed using the same steps as in the training phase.
  - Clearly, if a wrong prediction is made in one step, all the following steps will also be wrong.

Equipment setup

- The workstation was equipped with two Ethernet-based (NICs):
  - One for connecting it to the Internet,
  - The other one for connecting it to the Wi-Fi AP

- The workstation was configured to forward traffic between Wi-Fi AP and the Internet.

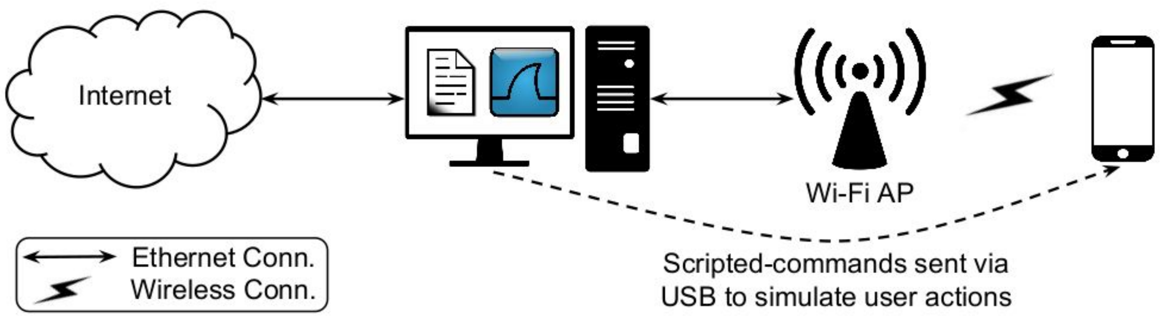
- The smartphones were provided (only one at a time) the Internet via Wi-Fi AP.



Equipment setup

- User actions were simulated on the smartphones using scripted-commands sent via USB or automation.
  - For Android, we used Android Debug Bridge (adb),
  - For iOS, we used Alloy 2.1.1 app that allows to automate the device without jailbreaking it.

- The generated network traffic was captured on the workstation using Wireshark 2.2.6; we discarded the packet's payload.



Evaluation

Evaluation settings

- We performed two different experiments:
  a. **Single classifier assessment:** in this setting, each single classifier is tested independently from the others.
  b. **Full stack classification:** in this setting, the classification is performed following the full-sequence of classification.

- We repeated each experiment **10 times** with a stratified **5-fold** cross validation on **90-10%** training and test splits of our dataset containing a total of **2362** instances.

Evaluation

Results - Single classifier assessment

Bitcoin *vs.* non-Bitcoin app classification

Method	Accuracy	Precision	Recall	F1
RF	0.977 ± 0.005•	0.977 ± 0.005•	0.973 ± 0.005•	0.975 ± 0.005•
SVM	0.930 ± 0.01	0.922 ± 0.01	0.923 ± 0.01	0.922 ± 0.02

App's OS classification

Method	Accuracy	Precision	Recall	F1
RF	0.984 ± 0.01•	0.984 ± 0.01•	0.983 ± 0.01•	0.983 ± 0.01•
SVM	0.956 ± 0.01	0.955 ± 0.02	0.955 ± 0.02	0.955 ± 0.02

Bitcoin app classification on Android

Method	Accuracy	Precision	Recall	F1
RF	0.966 ± 0.01•	0.968 ± 0.01•	0.968 ± 0.01•	0.968 ± 0.01•
SVM	0.945 ± 0.02	0.948 ± 0.02	0.948 ± 0.02	0.948 ± 0.02

Evaluation

Results - Single classifier assessment

Classification of user actions in Bitcoin apps on Android

Method	Bitcoin Wallet (Bitcoin.com)	BTC.com	Coinbase	Mycelium
RF	0.8 ± 0.15	0.98 ± 0.03•	0.991 ± 0.01•	0.971 ± 0.03•
SVM	0.85 ± 0.1•	0.975 ± 0.03	0.988 ± 0.02	0.958 ± 0.05

Bitcoin app classification on iOS

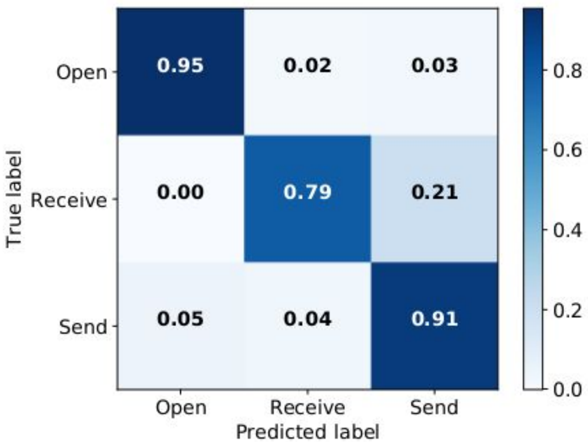
Method	Accuracy	Precision	Recall	F1
RF	0.962 ± 0.02•	0.964 ± 0.02•	0.963 ± 0.02•	0.963 ± 0.02•
SVM	0.935 ± 0.02	0.938 ± 0.02	0.935 ± 0.02	0.935 ± 0.02

Classification of user actions in Bitcoin apps on iOS

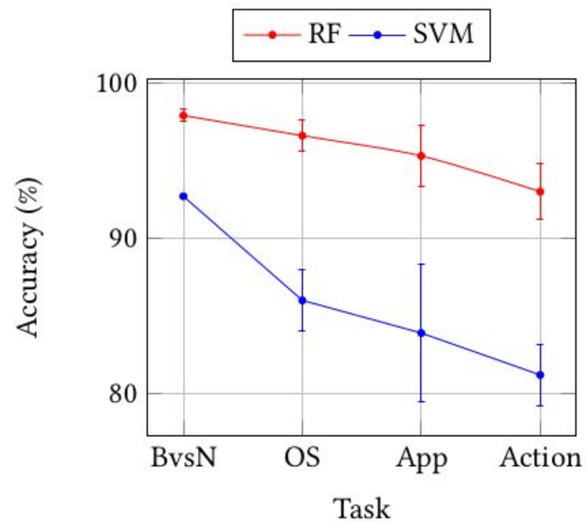
Method	Bitcoin Wallet (Bitcoin.com)	BitPay	Blockchain	Bread	Copay
RF	1.0 ± 0.0•	1.0 ± 0.0•	0.920 ± 0.02•	0.943 ± 0.03	1.0 ± 0.0•
SVM	1.0 ± 0.0•	1.0 ± 0.0•	0.911 ± 0.03	0.958 ± 0.04•	1.0 ± 0.0•

Evaluation

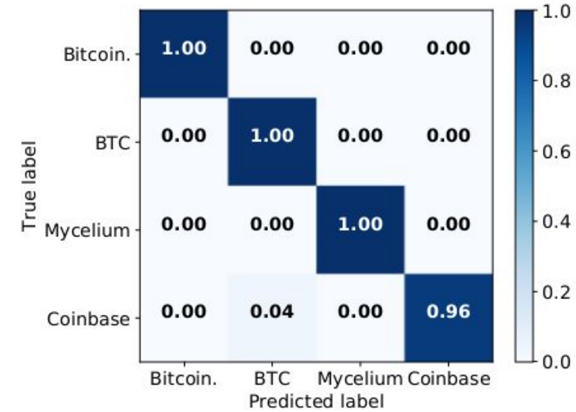
Results - Full stack classification



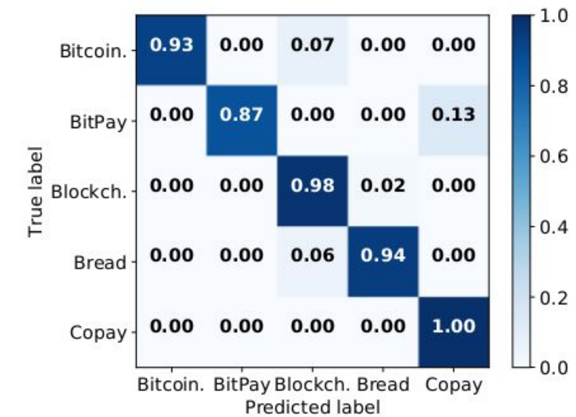
(a) Confusion Matrix



(b) Classification accuracy



(a) Android



(b) iOS

Future works

- In the future, we will investigate the security and privacy implication of transacting on such apps by considering a stronger adversary model.

- We will also explore the possibility to de-anonymize financial transaction placed via wallet apps.

References

4. M. Conti, A. Gangwal, and S. Ruj. 2018. “On the Economic Significance of Ransomware Campaigns: A Bitcoin Transactions Perspective,” in Elsevier Computers & Security 79 (2018), 162–189.

5. G. Ateniese, B. Hitaj, L. V. Mancini, N. V. Verde, and A. Villani. “No Place to Hide that Bytes won’t Reveal: Sniffing Location-Based Encrypted Traffic to Track a User’s Position,” in Springer Network & System Security 9408 (2015), 46–59.

6. X. Cai, X. Zhang, B. Joshi, and R. Johnson. “Touching From a Distance: Website Fingerprinting Attacks and Defenses,” in 19th ACM Computer and Communications Security (2012), 605–616.

7. Q. Wang, A. Yahyavi, B. Kemme, and W. He. “I Know What You Did on Your Smartphone: Inferring App Usage over Encrypted Data Traffic,” in 3rd IEEE Communications and Network Security (2015), 433–441.

8. M. Liberatore and B. Neil Levine. “Inferring the Source of Encrypted HTTP Connections,” in 13th ACM Computer and Communications Security (2006), 255–263.

9. V. N. Vapnik. “The Nature of Statistical Learning Theory,” Springer (1995) NY, USA.

Thank you!