AndroidアプリへのSQLインジェクション脆弱性検出を効率化するための自動ファジングテスト手法の提案

長崎県立大学情報セキュリティ学科　上田智浩

目次

- はじめに
- 関連研究
- 提案手法
- 評価
- 考察
- まとめ

はじめに

- スマートフォンが世界に広く普及し、Androidは2016年以降のモバイルOSのシェアで全体の7割を占めている
- アプリケーションに脆弱性がある場合、情報窃取や不正ログインといったサイバー攻撃を受ける可能性がある
- アプリケーションの脆弱性を自動で検査するツールは静的解析を実行するものが多く、既知の脆弱性を網羅的に検査できるが、未知の脆弱性は見つけにくい
- 未知の脆弱性の発見には動的解析が有効だが、動的にアプリケーションへの入力をテストするツールは少ない

➡

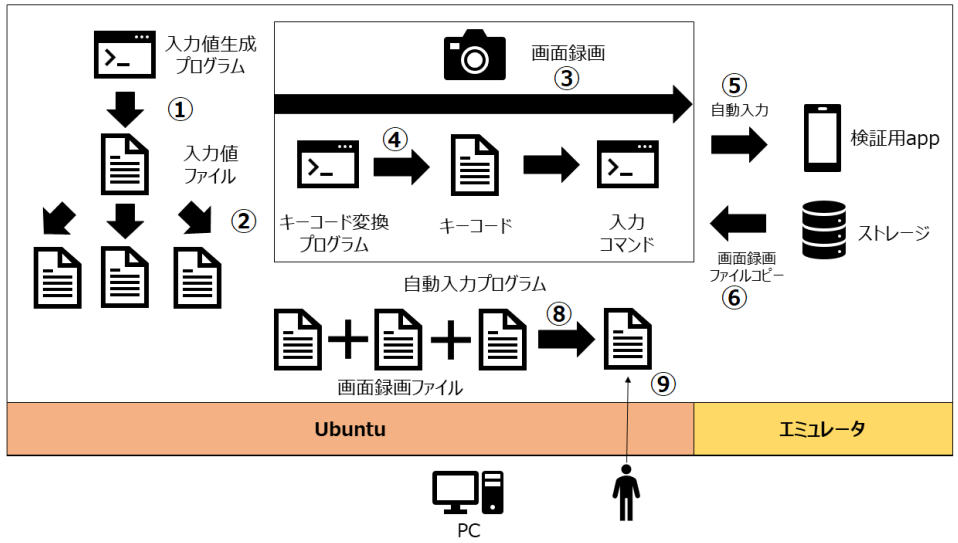
代表的なインジェクション攻撃であるSQLインジェクションの脆弱性を検査対象とし、動的にアプリケーションの入力欄をテストする

関連研究

- 佐久間らは既存のツールを用いて予測不可能なランダムな入力データを生成し、アプリケーションに自動入力することでエラーを発見するテスト方法を提案しているが、入力速度が1入力当たり2～3秒で実用的ではない
- Mohamedらは動的解析の効率を向上させるために、ランダムベースのツールと状態ベースのツールを組み合わせた入力生成方法を提案している
- Wangらはソースコードの静的解析から得たアプリケーションの脆弱性結果に基づいて動的解析の入力テストケースを構築し、入力テストの結果をテイント解析で検証する方法を提案している
- Aiminらは入力処理データに関するクラッシュバグを検出するために、様々な入力データのAPIを対象にした静的解析の結果に基づいてランダムな入力データを自動生成する方法を提案している
- HossainらはモバイルアプリケーションのセキュリティのSQLインジェクションの脆弱性について述べ、テイント解析を用いたSQLインジェクションの脆弱性を検出する方法を提案している

提案手法

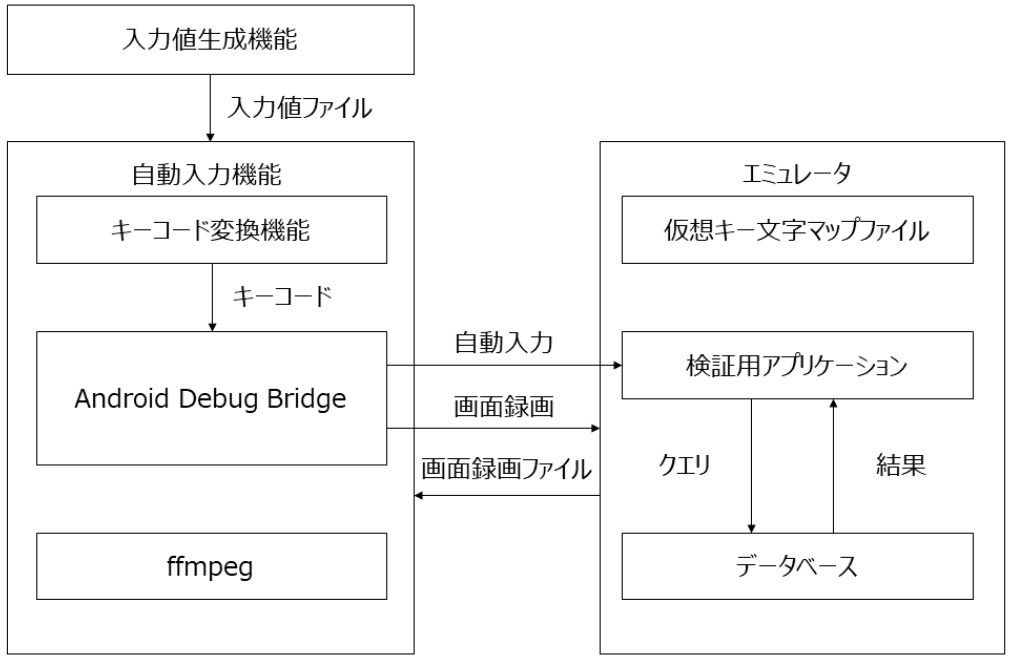
- SQLインジェクションの脆弱性を持つAndroidアプリケーションの入力欄に自動入力を実行し、脆弱性を検出する仕組みを提案する
- アプリケーションの画面に視覚的な変化がある攻撃を検知するために画面録画を実施する
- 自動入力と並行して画面録画することにより、攻撃が成功する入力値の特定を実現する



提案手法の概要

提案手法

◆コンポーネント



コンポーネントの関係図

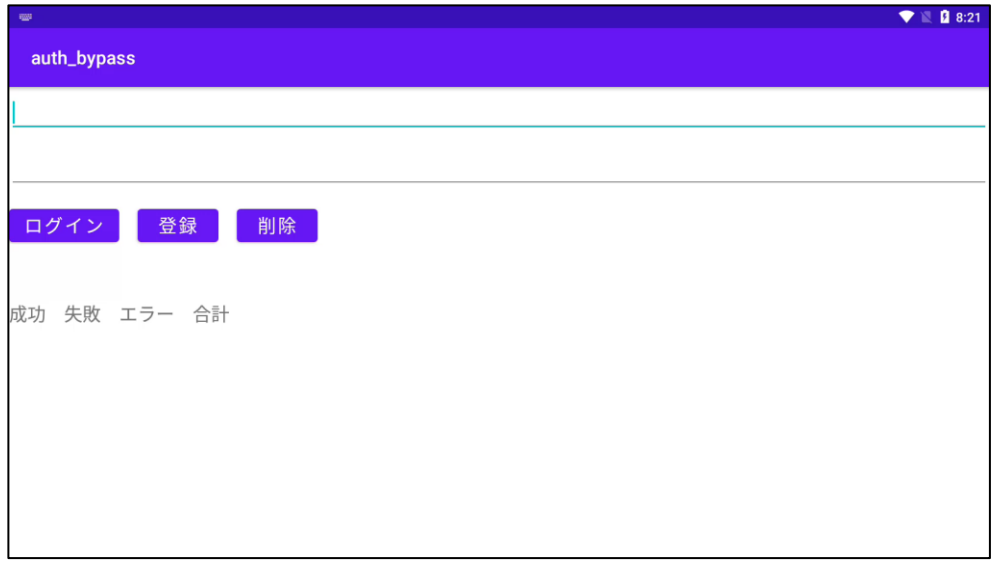
評価

- 入力速度、入力結果、画面録画結果の3項目で評価する
- 入力速度では画面録画なしの場合と画面録画ありの場合でそれぞれ10回ずつ計測し、平均値を算出する
- 入力結果では異なる入力値ファイルで10回入力し、成功、失敗、エラーのそれぞれの平均値を算出する
- 画面録画結果では画面録画ファイルのサイズと録画時間を示す

評価

◆入力速度

	画面録画なし	画面録画あり
1回目	13分44秒	18分52秒
2回目	13分40秒	16分39秒
3回目	13分44秒	16分57秒
4回目	13分47秒	17分23秒
5回目	13分51秒	17分13秒
6回目	14分9秒	16分56秒
7回目	12分54秒	16分25秒
8回目	12分51秒	17分19秒
9回目	12分58秒	16分48秒
10回目	12分51秒	17分10秒
平均値	13分26.9秒	17分10.2秒



自動入力の画面録画

評価

◆入力結果

	成功	失敗	エラー
1回目	131	4,028	5,841
2回目	120	4,088	5,792
3回目	144	4,153	5,703
4回目	147	4,133	5,720
5回目	138	4,077	5,785
6回目	124	3,967	5,909
7回目	136	4,091	5,773
8回目	127	4,120	5,773
9回目	148	4,046	5,806
10回目	126	4,079	5,795
平均値	134	4,078	5,787

自動入力終了時の画面

評価

◆画面録画結果

ファイル名 (.mp4)	サイズ	録画時間						
			result09	6.41MB	40秒	result18	6.20MB	38秒
result01	5.42MB	49秒	result10	6.26MB	42秒	result19	6.68MB	39秒
result02	6.32MB	41秒	result11	6.50MB	39秒	result20	6.42MB	39秒
result03	6.46MB	39秒	result12	6.42MB	38秒	result21	6.49MB	39秒
result04	6.70MB	41秒	result13	6.52MB	39秒	result22	6.38MB	38秒
result05	6.36MB	39秒	result14	6.32MB	38秒	result23	6.45MB	39秒
result06	6.36MB	39秒	result15	6.44MB	38秒	result24	6.47MB	38秒
result07	6.30MB	40秒	result16	6.33MB	38秒	result25	6.48MB	39秒
result08	6.32MB	38秒	result17	6.50MB	38秒	result	159MB	16分37秒

考察

◆入力速度

- 1入力当たりの入力速度は0.08～0.10秒であり、関連研究では1入力当たり2～3秒かかるため、実用的な入力速度に近づいたと考える
- adbで実現している機能を他のツ ールで代用できた場合、inputコマンドやscreenrecordコマンドの制約がなくなるため、より速くなると考える

◆入力結果

- ランダムに生成した入力値は攻撃に成功するものの精度が低い課題があり、機械学習やソースコードの静的解析を用いてテスト対象に合った入力値を生成することで攻撃成功率を高められると考える

まとめ

- Androidアプリの入力欄を動的にテストする自動入力システムを提案
- SQLインジェクションの脆弱性検出と攻撃に成功した入力値の特定を実現
- 関連研究と比べて速い入力速度を実現
- 機械学習を用いたより精度の高い入力値の生成や、画像処理による攻撃に成功した入力値の特定自動化を目指す