



FINANTEQ®

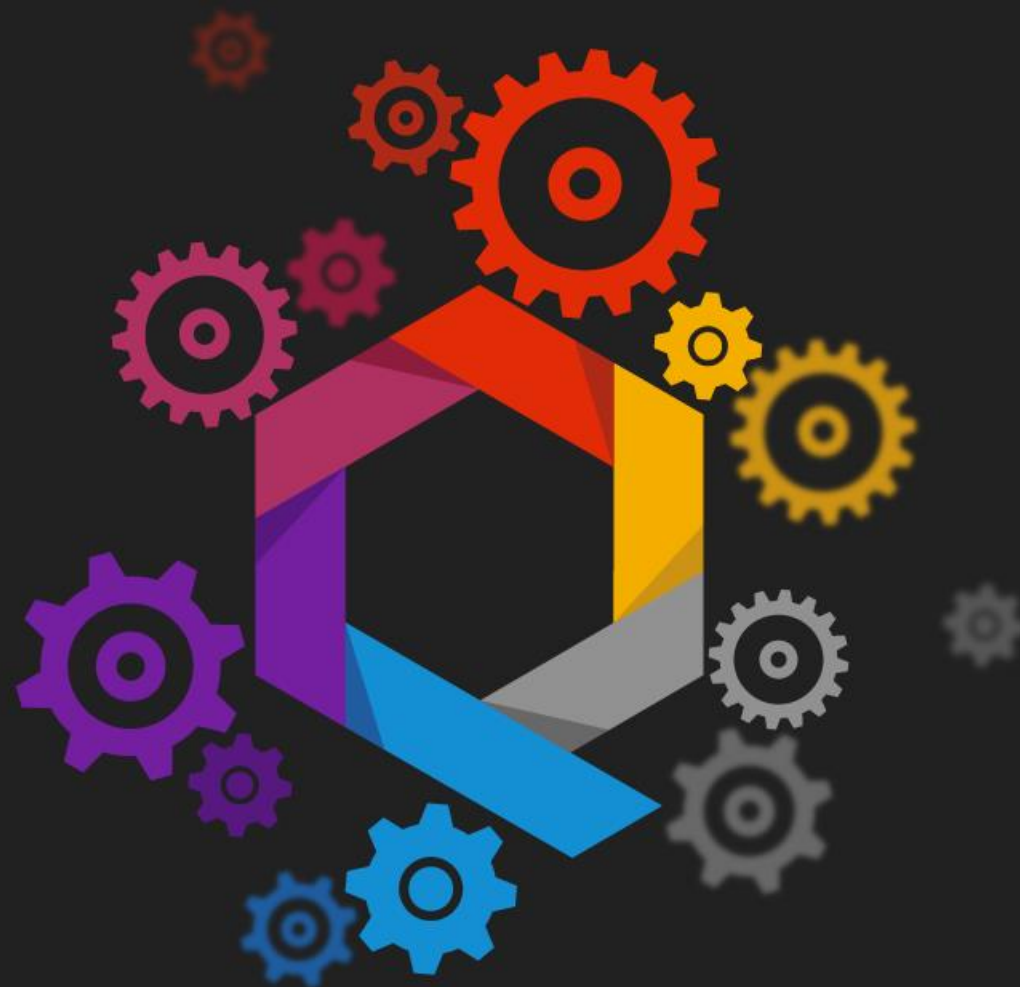


Android security tips & tricks

Autor Kamil Kalisz

O czym porozmawiamy ?

Analiza pamięci
Przegląd danych aplikacji
Modyfikacja pamięci



Co musimy wiedzieć

- Root!
- Cele ataków oraz możliwe korzyści
- Kim jest nasz „wróg”
- Świadomość



Analiza pamięci

- Stosunkowa łatwość wykonania
- Dużo podatnych aplikacji
- Szybka weryfikacja bezpiecznego przechowywania danych
- Możliwość łatwej ekstrakcji wrażliwych danych

Analiza pamięci

Wymagania :

- Root
- MAT (Eclipse Memory Analyzer)
- Android Device Monitor / adb
- Hprof-conv

Zabezpieczenia :

- Własne metody wprowadzania
- Szyfrowanie danych w pamięci
- SecureString

Analiza pamięci wykonanie

- `adb shell am dumpheap com.mypackage
/data/local/tmp/mypackage.hprof`
- `adb pull /data/local/tmp/mypackage.hprof`
- `hprof-conv mypackge.hprof mypackage.fixed.hprof`
- Plik wynikowy możemy otworzyć w Eclipse Memory Analyser (MAT)

Przegląd danych aplikacji

- Bardzo proste do wykonania
- Możliwość pozyskania „pozornie” nieistotnych danych: email, numer telefonu
- Jak i bardzo istotnych danych: loginy, hasła

Przegląd danych aplikacji

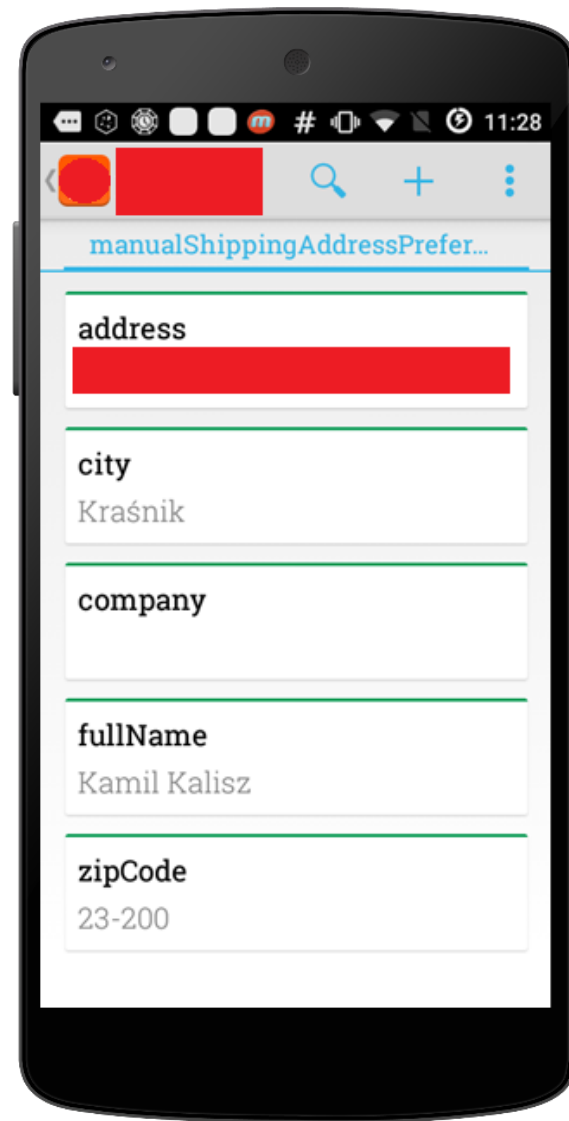
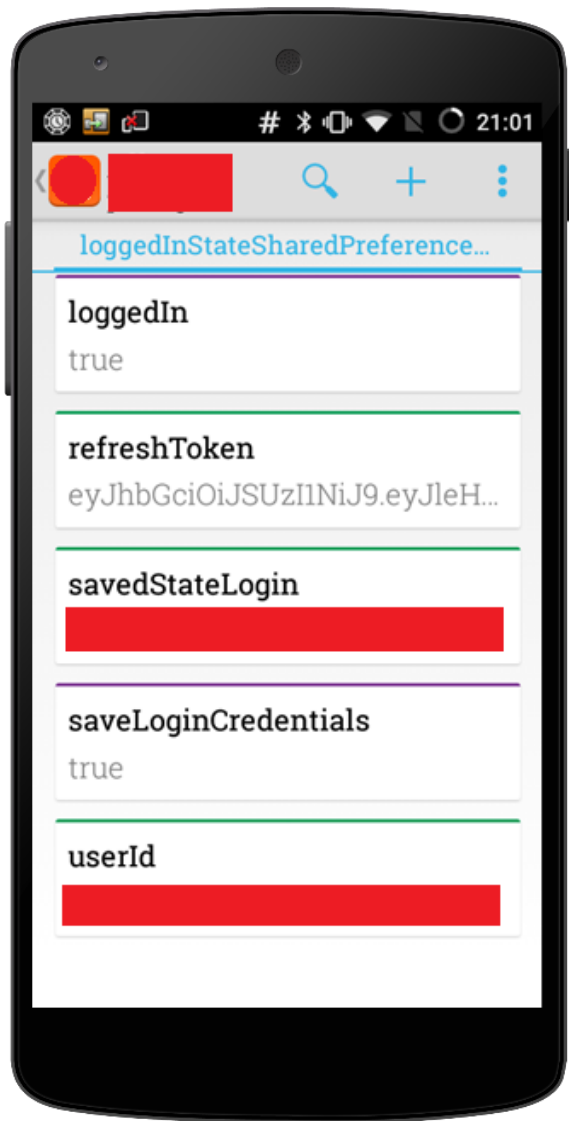
Wymagania :

- Root
- Preference Manager
- SQLite Editor

Zabezpieczenia :

- Szyfrowanie preferencji
- Szyfrowana baza danych
- Szyfrowane pliki tymczasowe

Alle.. dane



Modyfikacja pamięci

- W pewnych warunkach bardzo łatwa do wykonania
- Zależnie od aplikacji może się okazać, że jest to jedyny wektor ataku, którego musimy użyć

Modyfikacja pamięci

Narzędzia :

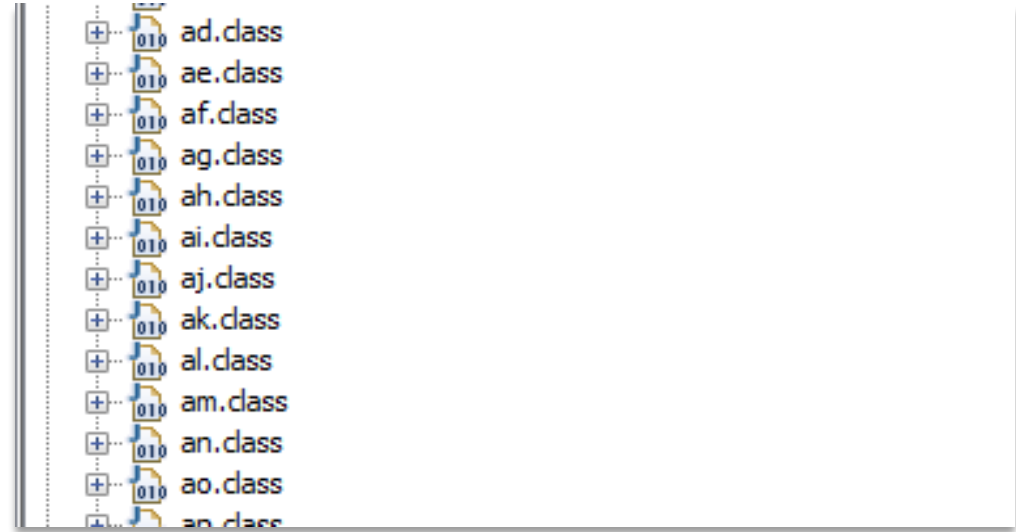
- Narzędzia typu GameKiller
- Własny natywny kod





Zabezpieczenia :

- Szyfrowanie danych w pamięci
- Weryfikacja danych przy dostępie

Weapon Throwing RPG

- Obfuscacja
- Szyfrowana baza danych
- Czy to wystarczy!?



	libandengine.so	2013-05-16 20:18
	libdatabase_sqlcipher.so	2013-06-27 11:58
	libsqlcipher_android.so	2013-06-27 11:58
	libstdlport_shared.so	2013-06-27 11:58

```
private long lifeCounter = 0;
private TamperSecured manaCounter = new TamperSecured(0);

private TextView manaCounterView;
private TextView lifeCounterView;

} private void increaseLife() {
    long newLifeValue = lifeCounter + random.nextInt(MAX_STATISTIC_PROGRESS);
    lifeCounter = newLifeValue;
    lifeCounterView.setText("Life: " + lifeCounter);
}

} private void increaseMana() {
    long newManaValue = manaCounter.getValue() + random.nextInt(MAX_STATISTIC_PROGRESS);
    manaCounter.setValue(newManaValue);
    manaCounterView.setText("Mana: " + manaCounter.getValue());
}

}
```

```

public class TamperSecured {

    private byte[] hash = new byte[0];
    private long value;

    public TamperSecured(long value) {
        this.value = value;
        hash = Sha256Utils.digest(value);
    }

    public long getValue() {
        validateValue();
        return value;
    }

    private void validateValue() {
        if (!Arrays.equals(hash, Sha256Utils.digest(value))) {
            throw new IllegalStateException("Memory tampered");
        }
    }

    public void setValue(long value) {
        validateValue();
        hash = Sha256Utils.digest(value);
        this.value = value;
    }
}

```

Rozwiązanie :

- Przy zapisie sprawdzanie skrótu aktualnej wartości, ustawianie zmiennej oraz liczenie skrótu nowej wartości
- Przy odczycie sprawdzanie skrótu aktualnej wartości

Dziękuję



FINANTEQ®

Android security tips & tricks

<https://github.com/kkalisz/gdglublin-android-security>

<http://finanteq.com>

kariera@finanteq.com

Narzędzia wykorzystane podczas przygotowań i prelekcji

- [AirDroid](#)
- [Preference Manager](#)
- [SQLite Editor](#)
- [Eclipse Memory Analyzer](#)
- [Towelroot](#)
- [Kingroot](#)
- [GameKiller](#)