SECURITY CASE 2015 STUDY

APLIKACJE ANDROID – NIEBEZPIECZNE PRAKTYKI

Dawid Pachowski, Michał Szklarski, Patryk Tenderenda

O NAS

rivet group

KN
Koło Naukowe Informatyków

security rivet group

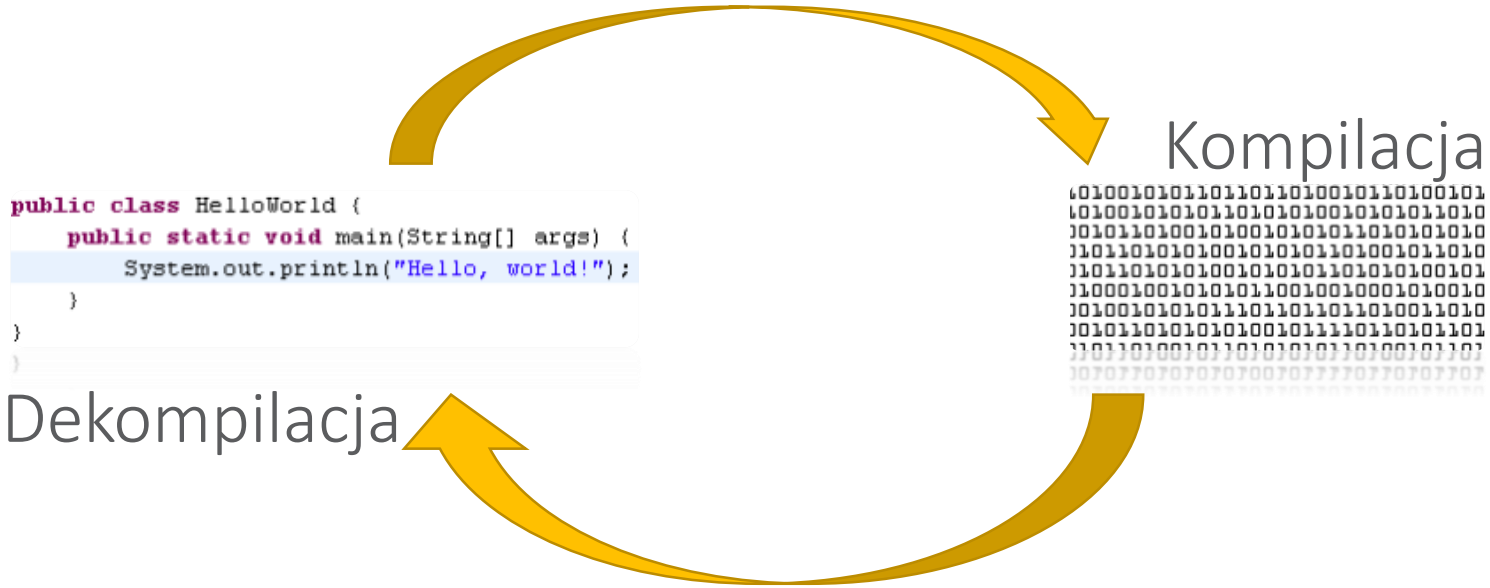
SCS 2015

PLAN PREZENTACJI

1. Definicje i przykłady
2. Prezentacja aplikacji
3. Dekompilacja
4. Analiza uzyskanego kodu
5. Podsumowanie

DEFINICJE I PRZYKŁADY

KOMPILACJA I DEKOMPILACJA



Dekompilacja jest to próba uzyskania kodu źródłowego z pliku binarnego aplikacji

DEKOMPILACJA APLIKACJI ANDROIDOWEJ

- Android = XML + Java
  - XML
    - Odpowiada za treść i wygląd w widokach oraz ustawienia
  - Java – kod pośredni
    - Odpowiada za logikę aplikacji
    - Android API

DEKOMPILACJA APLIKACJI ANDROIDOWEJ

Oryginał

```
package com.example.cryptopasscompare;

import android.os.Bundle;

public class SecondActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.second, menu);
        return true;
    }

}
```

Zdekompilowany

```
package com.example.cryptopasscompare;

import android.app.Activity;

public class SecondActivity extends Activity {

    protected void onCreate(Bundle paramBundle) {
        super.onCreate(paramBundle);
        setContentView(2130903041);
    }

    public boolean onCreateOptionsMenu(Menu paramMenu) {
        getMenuInflater().inflate(2131165185, paramMenu);
        return true;
    }
}
```

OBFUSKACJA W ANDROIDZIE

- Obfuskacja – zaciemnianie kodu

```
public void onClick(final View view) {
    Log.e("MYTAG", new String(MainActivity.this.encrypt("zxcvbnm", EditText.this.getText().toString())));
    if (MainActivity.this.p.equals(new String(MainActivity.this.encrypt("zxcvbnm", EditText.this.getText().toString())))) {
        MainActivity.this.startActivity(new Intent(MainActivity.this.getApplicationContext(), (Class)SecondActivity.class));
        MainActivity.this.finish();
    }
    else {
        Toast.makeText(MainActivity.this.getApplicationContext(), (CharSequence)"Wrong pass", 1).show();
    }
}
```

```
public void onClick(final View view) {
    if (this.a.a.equals(new String(this.a.a("zxcvbnm", this.b.getText().toString())))) {
        this.a.startActivity(new Intent(this.a.getApplicationContext(), (Class)SecondActivity.class));
        this.a.finish();
    }
    else {
        Toast.makeText(this.a.getApplicationContext(), (CharSequence)"Wrong pass", 1).show();
    }
}
```

OBFUSKACJA W ANDROIDZIE

- Obfuskacja – zaciemnianie kodu

```
public void onClick(final View view) {
    Log.e("MYTAG", new String(MainActivity.this.encrypt("zxcvbnm", EditText.this.getText().toString())));
    if (MainActivity.this.p.equals(new String(MainActivity.this.encrypt("zxcvbnm", EditText.this.getText().toString())))) {
        MainActivity.this.startActivity(new Intent(MainActivity.this.getApplicationContext(), (Class)SecondActivity.class));
        MainActivity.this.finish();
    }
    else {
        Toast.makeText(MainActivity.this.getApplicationContext(), (CharSequence)"Wrong pass", 1).show();
    }
}
```

```
public void onClick(final View view) {
    if (this.a.a.equals(new String(this.a.a("zxcvbnm", this.b.getText().toString())))) {
        this.a.startActivity(new Intent(this.a.getApplicationContext(), (Class)SecondActivity.class));
        this.a.finish();
    }
    else {
        Toast.makeText(this.a.getApplicationContext(), (CharSequence)"Wrong pass", 1).show();
    }
}
```

Zamiast *MainActivity* uzyskaliśmy *a* oraz zamiast *p* utrzymaliśmy *b*

Zamiast *encrypt* uzyskaliśmy *a*

Zamiast *EditText* uzyskaliśmy *b*

PREZENTACJA APLIKACJI

DEKOMPILACJA

SCS 2015

ANALIZA UZYSKANEGO KODU

PODSUMOWANIE

PODSUMOWANIE – ZNALEZIONE BŁĘDY

1.	SQL Injection
2.	Niezabezpieczone Activity
3.	Zahardcodowane ważne zmienne

PODSUMOWANIE – ZNALEZIONE BŁĘDY

4.	Słabe algorytmy kryptograficzne (MD5)
5.	Niepotrzebne uprawnienia
6.	Użyta funkcja skrótu bez soli

PODSUMOWANIE – CWE TOP 25

1.	SQL Injection	CWE-89: 'SQL Injection‘
2.	Niezabezpieczone Activity	CWE-306: Missing Authentication for Critical Function
3.	Zahardcodowane ważne zmienne	CWE-798: Use of Hard-coded Credentials

PODSUMOWANIE – CWE TOP 25

4.	Słabe algorytmy kryptograficzne (MD5)	CWE-327: Use of a Broken Risky Cryptographic Algorithm
5.	Niepotrzebne uprawnienia	CWE-250: Execution with Unnecessary Privileges
6.	Użyta funkcja skrótu bez soli	CWE-759: Use of a One-Way Hash without a Salt

PODSUMOWANIE – CWE

7.	Włączony tryb debug i logowanie
8.	Słaba obfuskacja
9.	Brak zależności danych podawanych od usera i kluczy do hashowania

UŻYTE KOMENDY - ADB

- Pobranie listy zainstalowanych aplikacji:
  adb shell pm list packages
- Pobranie informacji o aplikacji:
  adb shell dumpsys package
  eu.rivetgroup.security.cryptonotepad
- Ścieżka do pliku apk
  adb shell pm path eu.rivetgroup.security.cryptonotepad
- Pobranie pliku apk.
  adb pull /data/app/eu.rivetgroup.security.cryptonotepad-1.apk

UŻYTE KOMENDY - ADB

- Zmiana uprawnień do pliku (bazy danych) na 666 (read + write dla każdego)
  adb shell "run-as eu.rivetgroup.security.cryptonotepad chmod 666 /data/data/eu.rivetgroup.security.cryptonotepad/databases/cryptoNotes"
- Pobranie bazy danych
  adb pull /data/data/eu.rivetgroup.security.cryptonotepad/databases/cryptoNotes

UŻYTE KOMENDY - ADB

- Przekierowanie portu z lokalnego komputera na urządzeniu z Androidem
  adb forward tcp:31415 tcp:31415
- Uruchomienie aktywności z dodatkowymi parametrami
  adb shell am start -n eu.rivetgroup.security.cryptonotepad/eu.rivetgroup.security.cryptonotepad.activities.CryptoNotesList --ei "userId" 1

UŻYTE KOMENDY - DROZER

- Uruchomienie konsoli
  drozer console connect
- Wyszukanie aplikacji na liście wszystkich aplikacji
  run app.package.list -f cryptonotepad
- Wyświetlenie informacji o aplikacji
  run app.package.info -a
  eu.rivetgroup.security.cryptonotepad
- Przeskanowanie aplikacji pod względem podatności
  run app.package.attacksurface
  eu.rivetgroup.security.cryptonotepad

UŻYTE KOMENDY - DROZER

- Wyświetlenie wyeksportowanych aktywności
  run app.activity.info -a eu.rivetgroup.security.cryptonotepad
- Uruchomienie aktywności
  run app.activity.start --component
  eu.rivetgroup.security.cryptonotepad
  eu.rivetgroup.security.cryptonotepad.activities.CryptoNotesList
- Uruchomienie aktywności z parametrami
  run app.activity.start --component
  eu.rivetgroup.security.cryptonotepad
  eu.rivetgroup.security.cryptonotepad.activities.CryptoNotesList
  --extra integer userId 1

PYTANIA?

https://github.com/RivetSecurity/CryptoNotepad

DZIĘKUJEMY

SCS 2015