

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО»**

Факультет безопасности информационных технологий

Дисциплина:

«Обеспечение информационной безопасности мобильных устройств»

ОТЧЕТ ПО ЛАБАРАТОРНОЙ РАБОТЕ №3

«Основы анализа защищенности мобильных приложений»

Выполнил:

Гаврилова В. В., студент группы N33471


(подпись)

Проверил:

Федоров Иван Романович

24.02.2022

(отметка о выполнении)

(подпись)

Санкт-Петербург

2022г.

Содержание

1. Цель работы

2. Основная часть

3. Выводы

1. Цель работы

Ознакомиться с Owasp Mobile Top 10, на практике изучить возможные уязвимости мобильных приложений при помощи приложения Diva и моделирования в Genymotion.

2. Основная часть

2.1 Описание выбранных средств реализации и обоснования выбора

Для эмулярования был использован Genymotion, эмулировался Android 4.4 на Google Nexus 5. Для изучения уязвимостей был установлен diva-beta.apk.

2.2 Сборка стенда для пентестинга

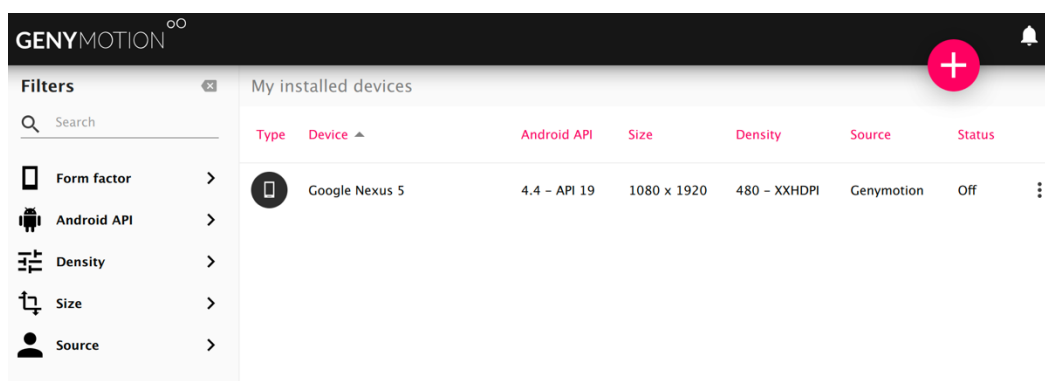


Рисунок 1 – Стенд в Genymotion

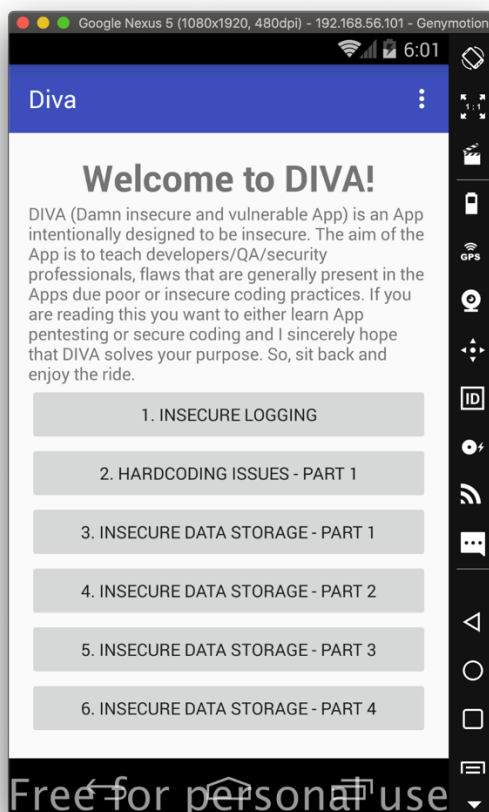


Рисунок 2 – Diva на Nexus 5

```
MBP-Veronika:~ veronikagavrilova$ adb devices
List of devices attached
192.168.56.101:5555    device

MBP-Veronika:~ veronikagavrilova$ adb connect 192.168.56.101
already connected to 192.168.56.101:5555
MBP-Veronika:~ veronikagavrilova$ adb logcat
```

Рисунок 3 – Подключенное устройство

2.3 Insecure logging

Цель эксплуатации уязвимости – выяснить пароль, введенный пользователем. Часто приложения для Android записывают конфиденциальную информацию в logcat.

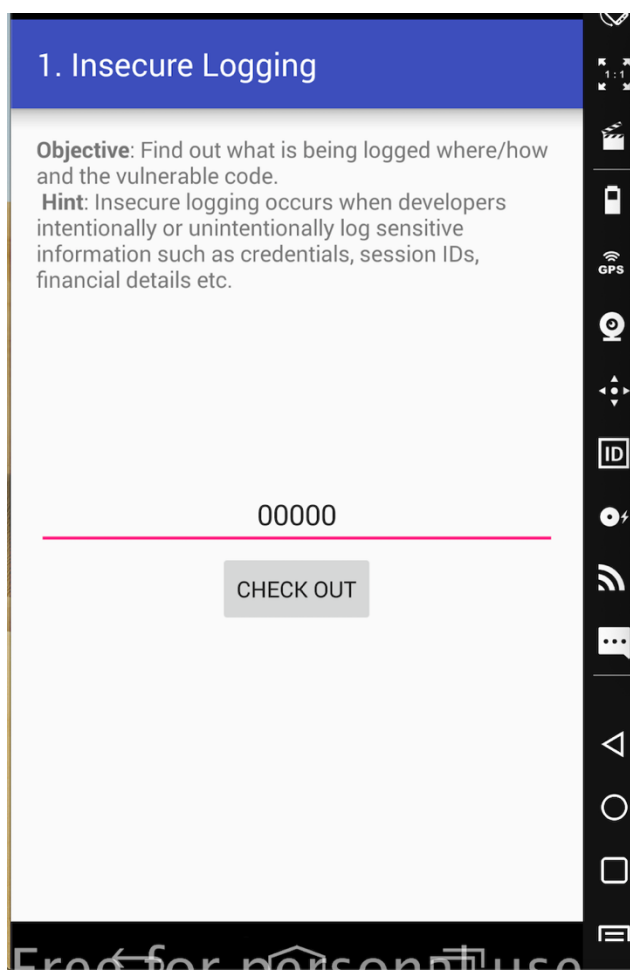


Рисунок 4 – Ввод пароля

```
MBP-Veronika:~ veronikagavrilova$ adb shell
root@vbox86p:/ # ps | grep diva
u0_a59    4853   1036   573712 47412 ffffffff b765407b S jakhar.aseem.diva
root@vbox86p:/ # logcat | grep 4853
D/dalvikvm( 4853): Late-enabling CheckJNI
```

Рисунок 5 – Просмотр логов, относящихся к процессам Diva

```
E/EGL_emulation( 4853): tid 4853: eglSurfaceAttrib(1210): error 0x3009 (EGL_BAD_MATCH)
W/HardwareRenderer( 4853): Backbuffer cannot be preserved
E/diva-log( 4853): Error while processing transaction with credit card: 0000000000
D/dalvikvm( 4853): Debugger has detached; object registry had 1 entries
```

Рисунок 5 – Пароль

2.4 Hardcoding issues – part 1

Цель – узнать пароль производителя, чтобы получить доступ. Разработчик может оставить пароль в нешифрованном виде в коде приложения.

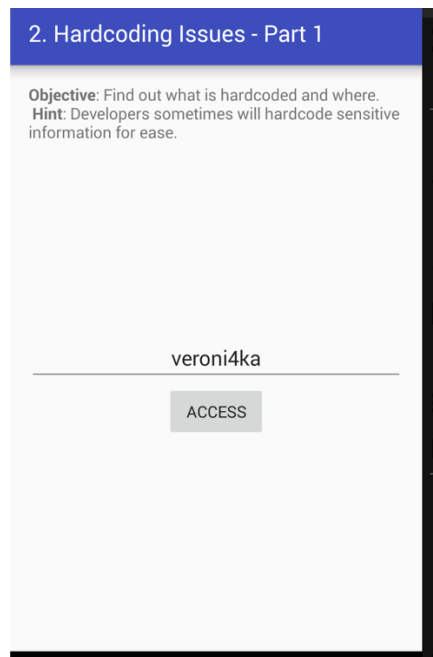


Рисунок 6 – Ввод пароля

```
HardcodeActivity.java
12  *
13  * This program is distributed in the hope that it will be useful,
14  * but WITHOUT ANY WARRANTY; without even the implied warranty of
15  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
16  * GNU General Public License for more details.
17  *
18  * You should have received a copy of the GNU General Public License
19  * along with this program. If not, see <http://www.gnu.org/licenses/>.
20  *
21  * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING,
22  * BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
23  * PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS
24  * BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
25  * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
26  * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
27  * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
28  * OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
29  * POSSIBILITY OF SUCH DAMAGE.
30  *
31  */
32 package jakhar.aseem.diva;
33
34 import android.support.v7.app.AppCompatActivity;
35 import android.os.Bundle;
36 import android.view.View;
37 import android.widget.EditText;
38 import android.widget.Toast;
39
40 public class HardcodeActivity extends AppCompatActivity {
41     @Override
42     protected void onCreate(Bundle savedInstanceState) {
43         super.onCreate(savedInstanceState);
44         setContentView(R.layout.activity_hardcode);
45     }
46
47     public void access(View view) {
48         EditText hkey = (EditText) findViewById(R.id.hcKey);
49
50         if (hkey.getText().toString().equals("Vendorsecretkey")) {
51             Toast.makeText(this, "Access granted! See you on the other side :)", Toast.LENGTH_SHORT).show();
52         }
53         else {
54             Toast.makeText(this, "Access denied! See you in hell :D", Toast.LENGTH_SHORT).show();
55         }
56     }
57 }
58
```

Рисунок 7 – Содержимое HardcodeActivity.java и пароль в явном виде

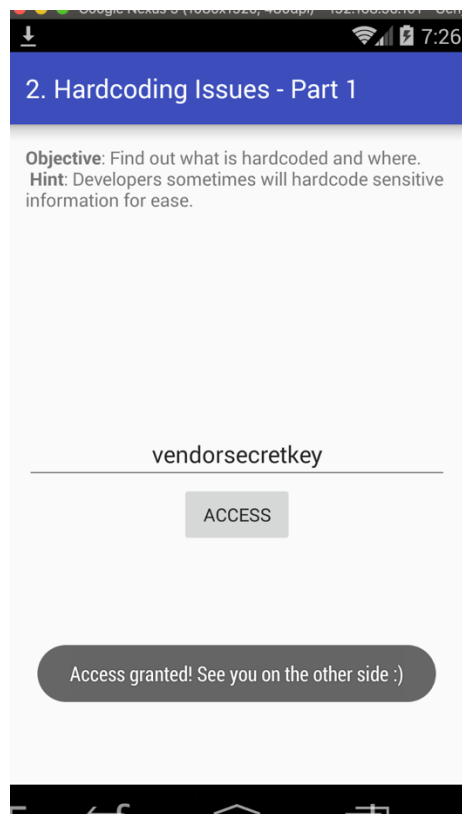


Рисунок 8 – Ввод найденного пароля

2.5 Input Validation Issues - Part I

После прочтения описания можно понять, что существует 3 пользователя. При помощи SQL инъекций можно выяснить логины и пароли пользователей.

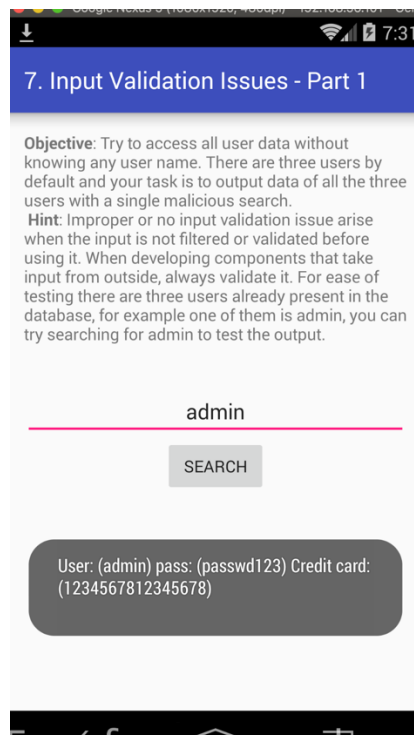


Рисунок 9 – Результат при вводе пользователя admin

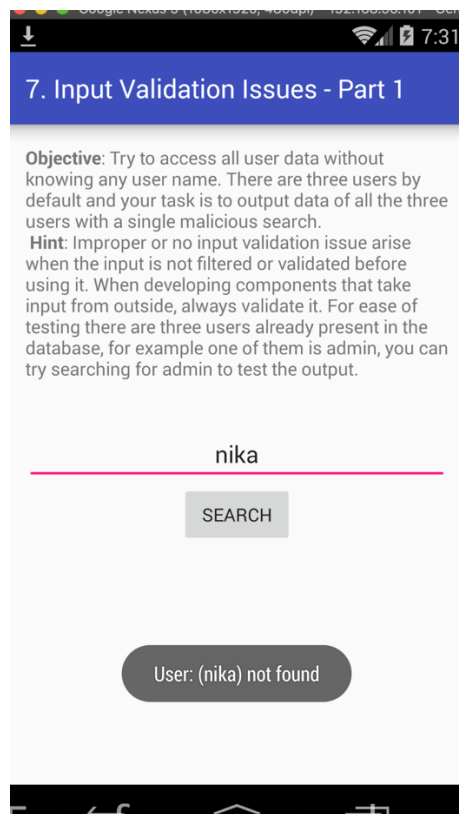


Рисунок 10 – Результат при вводе пользователя nika

Можно попробовать ввести в поле специальные символы, такие как \$ “ ‘ #

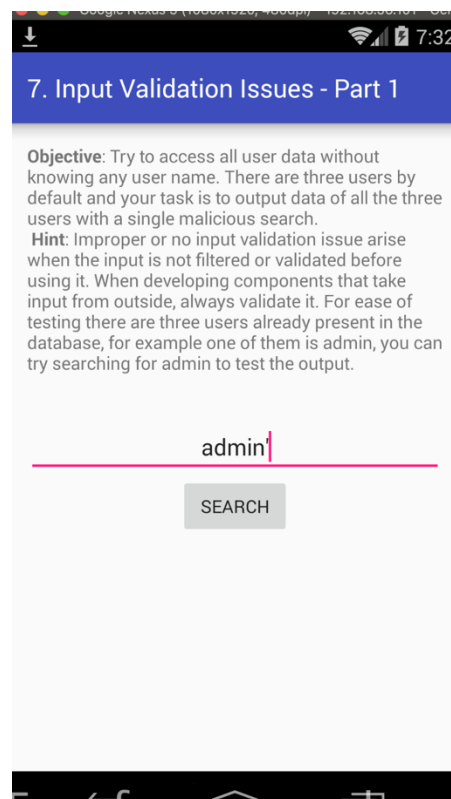


Рисунок 11 – Результат ввода специального символа – невидимая ошибка

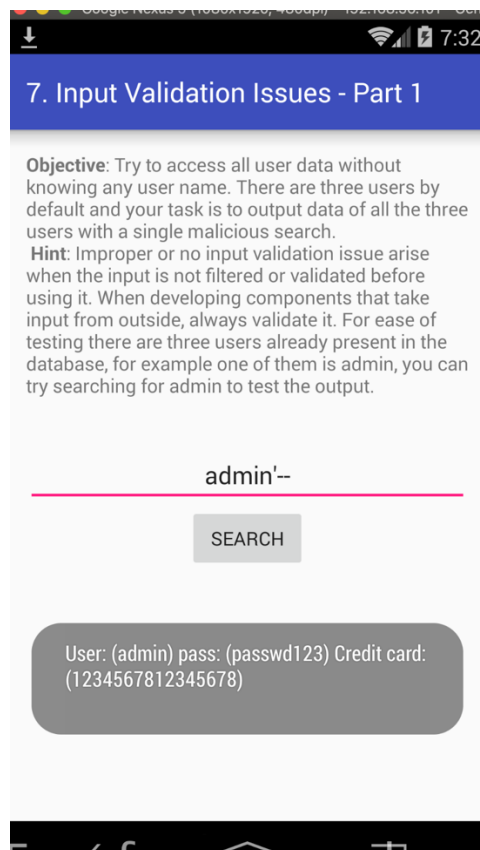


Рисунок 12 – Успешная попытка запроса с синтаксисом SQL

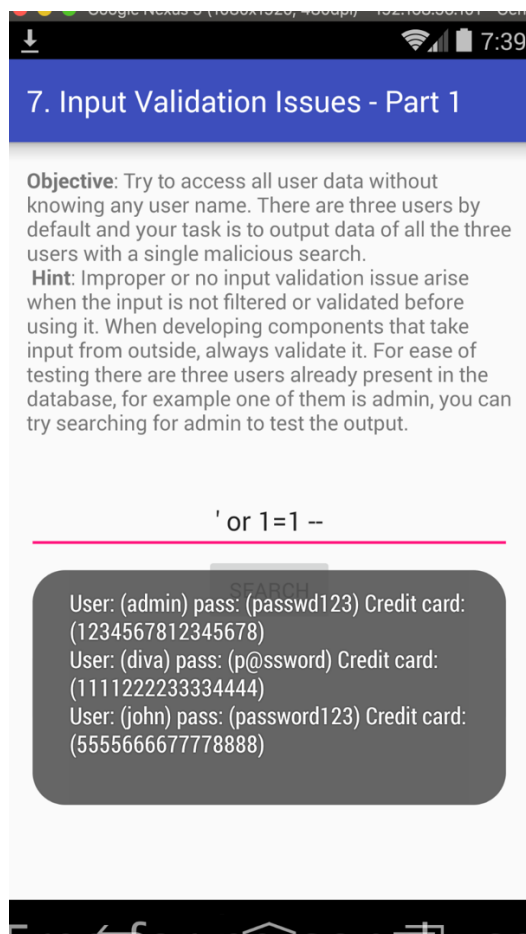


Рисунок 13 – Успешный запрос SELECT * FROM Table WHERE user="' or 1=1—'

3. Вывод

В результате выполнения работы я научилась настраивать и производить эмуляцию в Genymotion, изучила основные уязвимости в мобильных приложениях и проэксплуатировала их на практике в Diva.