Universidade Estadual do Norte Fluminense Darcy Ribeiro
CCT - Centro de Ciência e Tecnologia
LCMAT - Laboratório de Ciências Matemáticas
CC - Curso de Ciência da Computação

Relatório Anual

de Pesquisa Iniciação Científica PIBIC/CNPq

Novembro 2019 - Outubro 2020

Segurança em dispositivos móveis Android

José Lucio Castro Azevedo
Aluno Bolsista

Prof. Ausberto S. Castro Vera
Orientador

Campos dos Goytacazes, Outubro de 2020

1 Etapas propostas no plano de trabalho

No período —Novembro 2019 - Setembro 2020— foram desenvolvidas as seguintes atividades:

- Desenvolvedores treinados para o uso de ferramentas de desenvolvimento de software para dispositivos móveis
- Disponibilidade de uma base de informações sobre riscos e vulnerabilidades em dispositivos móveis.
- Desenvolvimento de uma aplicação relacionada com segurança
- Desenvolver a cultura dos dispositivos móveis no Curso de Ciência da Computação da UENF e incentivo a novos projetos sobre segurança computacional em dispositivos móveis

2 Introdução

O uso de dispositivos móveis, tanto pessoais quanto corporativos, é praticamente inevitável. Em muitos casos, eles são essenciais para para as pessoas, além de ser uma forma de passar o tempo. Por isso, a segurança móvel é um assunto em pauta e que recebe cada vez mais atenção na atualidade. Atualmente, grande parte da população mundial tem celular e fazem uso diário, de modo que, com o crescimento do seu uso, crescem também as ameaças [SRS20], [She20].

Os celulares viraram alvo de hackers e usuários mal-intencionados. Ameaças como phishing (tentativa fraudulenta de obter informações confidenciais como nomes de usuário), vírus e malwares (software mal-intencionado) atingem os dispositivos móveis. As falhas na gestão de segurança móvel podem levar a vazamento e perda de dados, espionagem, uso de aplicativos não autorizados e outros problemas [ALRR17].

[AA19] declara que *"os avanços tecnológicos na indústria de dispositivos móveis tornaram a vida mais flexível e próspera. Tudo pode ser feito por meio de smartphones que se tornaram parte da vida humana. No entanto, os riscos de segurança associados a esses dispositivos, como roubo e falsificação, estão aumentando e sua detecção continua sendo um desafio por vários motivos"*. Esse foi um dos desafios deste projeto de pesquisa: entender alguns conceitos e ferramentas para a deteccção de vulnerabilidades em dispositivos móveis.

3 Objetivos

O objetivo principal deste projeto foi pesquisar sobre o conjunto de riscos e vulnerabilidades nos dispositivos móveis e do sistema operacional Android e criar profissionais de segurança habilitados para o mercado. Para isto foi necessário elaborar uma base de informações sobre riscos, ameaças e vulnerabilidades, sobre implementações disponíveis e algoritmos a serem implementados.

4 Metodologia

As metodologias utilizadas na pesquisa, foram:

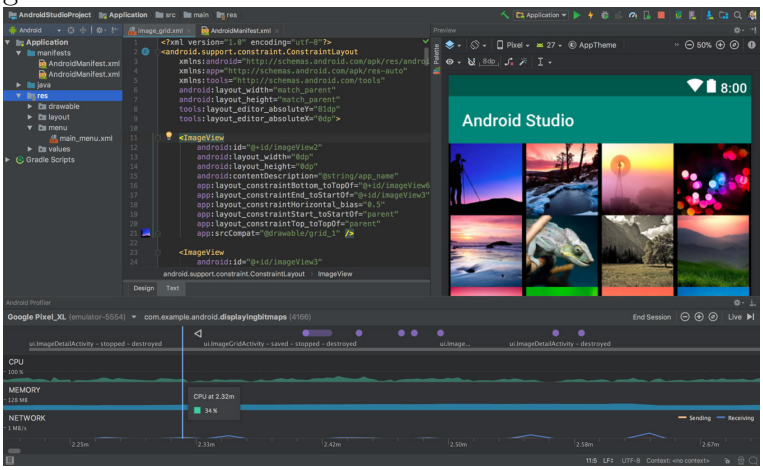
4.1 Pesquisas bibliográficas e Minicursos

Leitura de artigos científicos, Resumos, Livros e tccs. Minicursos como o "IT Security Specialist"criado por Daniel Donda, e alguns módulos do minicurso "Pentest Mobile"da empresa Desec Security, além de vários video-aulas sobre assuntos relacionados encontradas na internet.

4.2 Programação no Android Studio

O Android studio 3.6, é uma IDE (ambiente de desenvolvimento integrado) para desenvovimentos de aplicativos no sistema Android, é baseado no IntelliJ IDEA, foi utilizado para a criação de aplicativos [DDW16] e implementação de métodos de segurança.

Figura 1: Ambiente de desenvolvimento Android Studio 3.6



Além do editor de código e das ferramentas de desenvolvedor avançadas do IntelliJ, mostrado na figura 1, o Android Studio oferece ainda mais recursos para aumentar sua produtividade na compilação de apps Android, como:

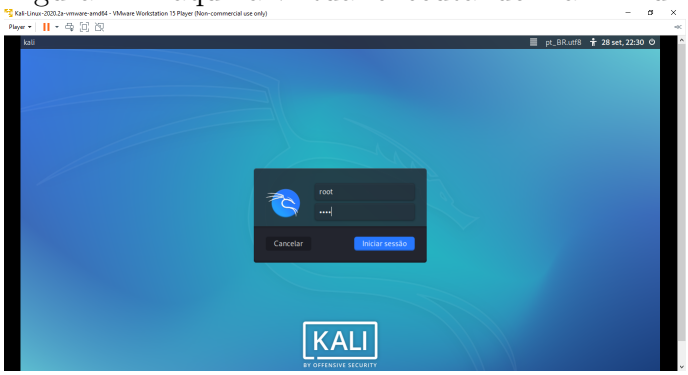
- Um emulador rápido com inúmeros recursos.
- Um sistema de compilação flexível baseado em Gradle.
- Um ambiente unificado que possibilita desenvolvimento para todos os dispositivos Android.
- Frameworks e ferramentas de teste cheios de possibilidades.

Tem como suas linguagens de programação Java e Kotlin.

4.3 Sistemas operacionais

Foram utilizados os sistemas operacionais Windons 10, como sistema principal, e o Kali Linux virtualizado [Wei14] (Figura 2) através da tecnologia vmware para a utilização de suas ferramentas de segurança muito reconhecidas no mercado.

Figura 2: Máquina virtual executando Kali Linux



4.4 Outras ferramentas utilizadas

Muitas outras tecnologias foram utilizadas ao decorrer do projeto, como:

- A proxy Charles, que é um servidor que age como um intermediário para requisições.
- O port scan nmap, um software livre que realiza o scan das portas TCP e UDP.
- A ferramenta de segurança Ettercap, usada para a realização de ataques man-in-the-middle.
- A ferramenta de segurança Ettercap, usada para a realização de ataques man-in-the-middle.

E algumas outras ferramentas de segurança encontradas no Kali Linux.

5 Resultados e Discussão

Para implementar conceitos de segurança, é importante primeiro aprender como fuciona a linguagem de programação utilizada e o processo de criação de um aplicativo. Durante o período da pesquisa, foram realizadas as seguintes atividades:

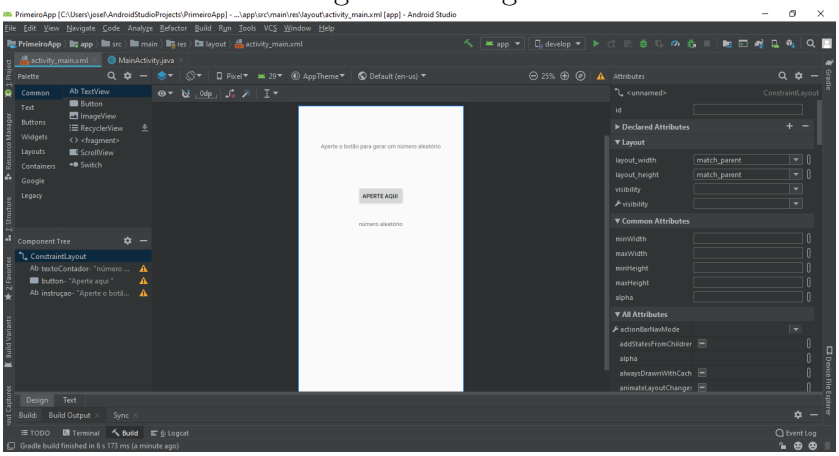
5.1 Linguagem Java

Java é uma linguagem de programação orientada a objetos desenvolvida na década de 90. Diferente das linguagens de programação mais modernas, que são compiladas para código nativo, a linguagem Java é compilada para um bytecode que é interpretado por uma máquina virtual. A linguagem Java foi projetada tendo em vista alguns objetivos. Entre eles estão, orientação a objetos, portabilidade, recursos de rede e segurança

5.2 Aplicativos Desenvolvidos

A seguir, apresentamos alguns dos aplicativos desenvolvidos, como parte das atividades da pesquisa e do aprendizado da linguagem Java:

Figura 3: Design

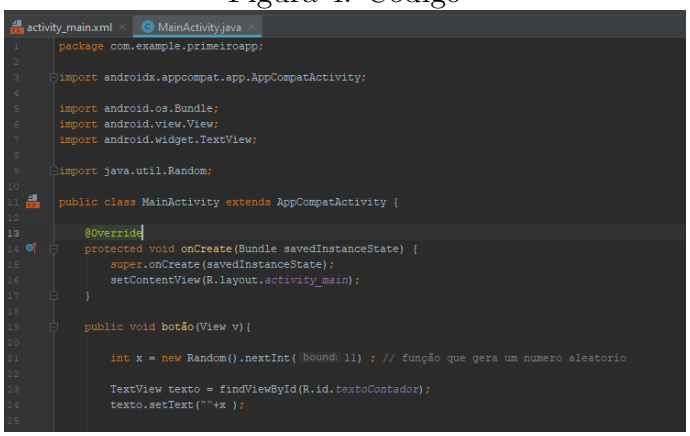


5.2.1 Gerador de números aleatórios

Um gerador de números aleatórios ativado por um botão, que atribuí o número ao textview com o comando setText

Ao apertar o botão visto na figura 3, ativa o código mostrado na figura 4 gera um número aleatório de 0 a 10 (limitado).

Figura 4: Código



```
package com.example.primeiroapp;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import java.util.Random;

public class MainActivity extends AppCompatActivity {

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

    public void botão(View v){

        int x = new Random().nextInt( bound: 11) ; // função que gera um numero aleatorio

        TextView texto = findViewById(R.id.textoContador);
        texto.setText(""+x );
```

5.2.2 Gerador de personagem aleatórios

Ao criar o aplicativo anterior se teve a ideia de implementar a seu algoritmo alguma coisa do dia a dia, no caso, personagens do jogo League of Legends. Foi colocado um Switch/Case (estrutura condicional) e os números de 1 a 10 em cada caso, e nesses casos era atribuído um personagem .

E por sua vez foi criada uma variável de entrada para receber uma string digitada pelo usuário referente a posição do personagem no jogo. Também foi criado outro Switch/Case e colocado o Switch/Case citado na figura 5 dentro dele. Tendo assim uma estrutura para dividir as posições de cada personagem baseado na string de entrada do usuário, e uma estrutura dentro da primeira selecionando os personagens com base no gerador de números aleatórios.

```
switch (x) {
    case 1:
        resultado.setText("Darius");
        imgresposta.setImageResource (R.drawable.darius);

        break;

    case 2:
        resultado.setText("Yorick");
        imgresposta.setImageResource (R.drawable.yorick);
        break;

    case 3:
        resultado.setText("Mordekaiser");
        imgresposta.setImageResource (R.drawable.morde);
        break;
```

Figura 5: estrutura condicional Switch/Case

5.2.3 Aplicativo de Quiz

Continuando com o aprendizado da linguagem Java, no Android Studio, se teve a ideia de criar um aplicativo que ao responder suas perguntar, ele calcula com quem você mais se parece do um grupo de amigos mostrado na figura 6. Um aplicativo um pouco mais trabalhoso, dessa vez com multiplos layouts ao invez de só um e com atribuição de uma foto com o comando setImageResource, não só uma string como vimos nos últimos 2 aplicativos.

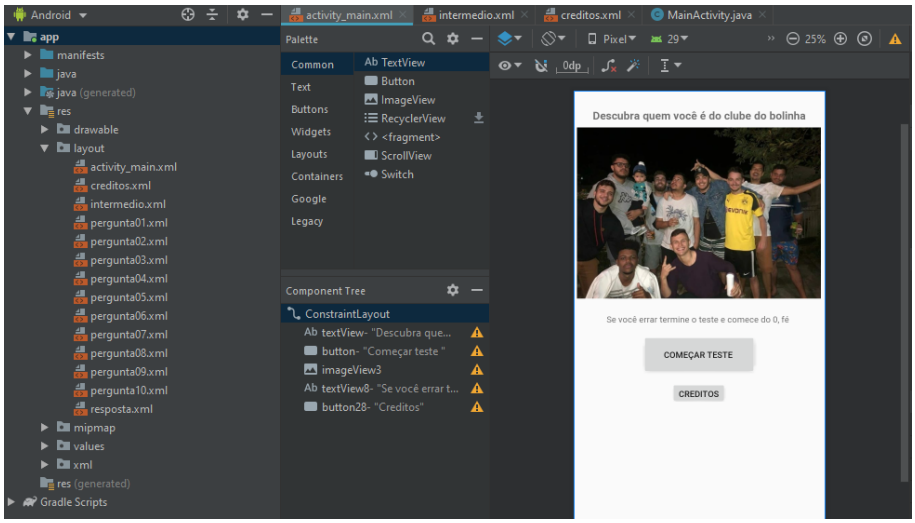


Figura 6: Quiz do Clube do Bolinha

A cada resposta se soma o valor 1 as variáveis correspondente a aquela resposta, depois é feita uma comparação de todas as variáveis. A foto e o nome do membro do grupo correspondente a variável de maior valor aparece na tela.

5.2.4 Aplicativo com senha

Para a proposta do quarto aplicativo surgiu o desafio do uso de uma senha para ter acesso ao conteúdo de um layout. O campo de entrada da senha foi colocado e a validação foi feito por meio de uma comparação simples. Mas com isso, a senha poderia ser vista no código fonte, assim a tornando vulnerável. A solução encontrada foi o uso de algoritmo de hash, que é um algoritmo unidirecional que mapeia dados de comprimento variável para dados de comprimento fixo. Os valores retornados por uma função hash são chamados

valores hash. o algoritmo escolhido foi o Md5, que é uma função que produz um valor de hash de 128 bits expresso em 32 caracteres.

```
private String convertPassMd5(String pass) {
    String password = null;
    MessageDigest mdEnc;
    try {
        mdEnc = MessageDigest.getInstance("MD5");
        mdEnc.update(pass.getBytes(), offset: 0, pass.length());
        pass = new BigInteger ( signum: 1, mdEnc.digest()).toString( radix: 16);
        while (pass.length() < 32) {
            pass = "0" + pass;
        }
        password = pass;
    } catch (NoSuchAlgorithmException e1) {
        e1.printStackTrace();
    }
    return password;
}


public void botaol(View view) {
    String senhazada = senha.getText().toString();
    md5 c1= new md5();
    senhazada = convertPassMd5(senhazada);


    switch (senhazada){
        case "0d39214e911f6b360af5a30a2a7b9e04": setContentView(R.layout.tela1);
        default: senhaerrada.setText("Senha errada");
    }
}
```

Figura 7: senha criptografada

A senha é digitada pelo usuário, passa pelo método convertPassMd5 que a criptografa em md5 e só depois é comparada com o Hash visto na figura 7.

5.2.5 Aplicativos de Criptografia

Após um estudo mais aprofundado sobre criptografia, seus tipos, como funcionam e como são quebradas percebesse que o algoritmo md5 já se mostra vulnerável entre os hashs mais atuais. Com isso o quinto aplicativo desenvolvido traz uma calculadora de hashs. Que apresenta a possibilidade da criação de um hash md5, mas também de algoritmos de hash mais seguros, como o Sha-1 e o Sha-256 (algoritmo usado na mineração do Bitcoin), como visto na figura 8.

Figura 8: Calculadora de Hashs

Já no sexto aplicativo criado foi implementado um algoritmo criptográfico de cifra simétrica.

A criptografia simétrica [Mor16] é usada para garantir a confidencialidade de determinado dado e garante que apenas os destinatários autorizados, aqueles que conheçam a chave de decifração, possam recuperar os dados originais. O algoritmo Advanced Encryption Standard, ou AES, é a cifra simétrica mais utilizada atualmente. É utilizado em muitas indústrias e várias aplicações, dentre as quais se pode citar o protocolo de segurança IP, o protocolo TLS, a encriptação de dispositivos WiFi (padrão IEEE 802.11i), o protocolo SSH (Secure Shell) e aplicativos muito famosos como Skype e WhatsApp.

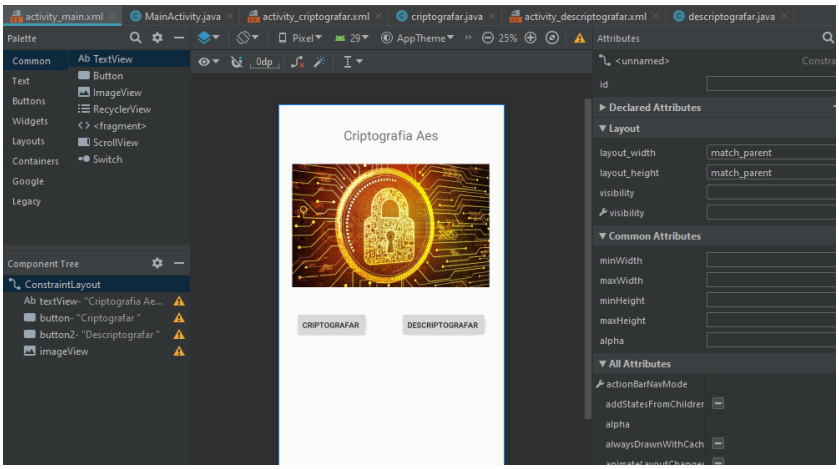


Figura 9: Design do sexto Aplicativo

O usuário entra com uma frase em texto plano (Caso queira criptografar) ou uma frase criptografada (Caso queira descriptografar) e com uma chave e o aplicativo exerce sua função. A escolha de criptografar ou descriptografar é mostrada na Fig.9.

5.2.6 Aplicativo de foto

Nesse aplicativo, foi necessário o estudo de permissões do Android Studio. O objetivo de uma permissão é proteger a privacidade do usuário. Aplicativos para Android precisam solicitar permissão para acessar dados confidenciais do usuário (como contatos e SMS), bem como recursos do sistema (como câmera e Internet). Dependendo do recurso, o sistema pode conceder a permissão automaticamente ou pedir ao usuário que aprove a solicitação. As permissões solititadas pelo aplicativo 7 foram as seguintes:

- Permissão para a leitura de dados armazenados:

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

- Permissões para o uso da câmera:

```
<uses-feature android:name="android.hardware.camera"android:required="true"/>
 <uses-permission android:name="android.permission.CAMERA" />
```

Com as permissões mais o código devido, é possivel por meio desse aplicativo tirar uma foto ou procurá-la em sua galeria para exibir na tela como mostrada na Fig.10. O aplicativo teve alguns problemas com o uso dos recursos de câmera, mas o estudo das permissões foi de grande importância.

Figura 10: Aplicativo de câmera

5.3 Malwares

Malwares [CGI12] são programas desenvolvidos para executar de atividades maliciosas em um computador, pela exploração de vulnerabilidades, auto-execução de mídias removíveis infectadas(pen-drives), pelo acesso a páginas Web maliciosas, pela ação direta de atacantes que, após invadirem o computador, incluem arquivos contendo códigos malicioso, etc. Uma vez instalados, os códigos maliciosos passam a ter acesso aos dados armazenados no computador e podem executar ações em nome dos usuários, de acordo com as permissões de cada usuário.

5.4 Ataques

Para a familiarização com ataques hackers conhecidos, o projeto não se conteve a teoria. Foram realizados alguns ataques conhecidos [Mor17] em ambientes de teste controlados e em dispositivos com permissão dos usuários.

5.4.1 API Cielo eCommerce

Para entender o funcionamento de uma API de pagamento para realização do resumo do confict, o teste da API Cielo eCommerce teve grande importância. Ao receber a dica de que talvez o servidor para testes não funcionava da forma apropriada, surgiu o interesse de testar essa aplicação. O servidor de testes funcionava baseado em http, isso é, as requisições seriam enviadas em texto plano, assim vazando as informações sensíveis do usuário (sem problemas por ser um servidor apenas para testes). Para comprovar o fato, foi feita uma requisição nesse servidor pelo Postman (plataforma de teste de APIs).

Após a requisição mostrada na Fig.11, foi feito um pequeno Man-in-the-middle, interceptando as informações requisitadas pelo postman com a proxy Charles. Ao checar essas informações ficou comprovado que elas eram enviadas sem criptografia (texto plano) pelo servidor de testes, como podemos ver na Fig.12.

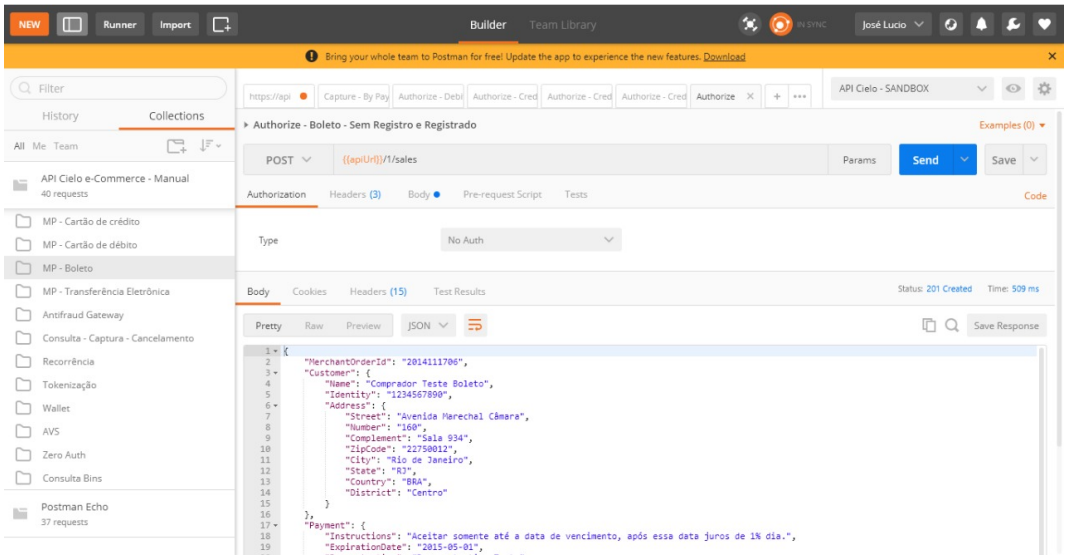


Figura 11: Requisição de dados

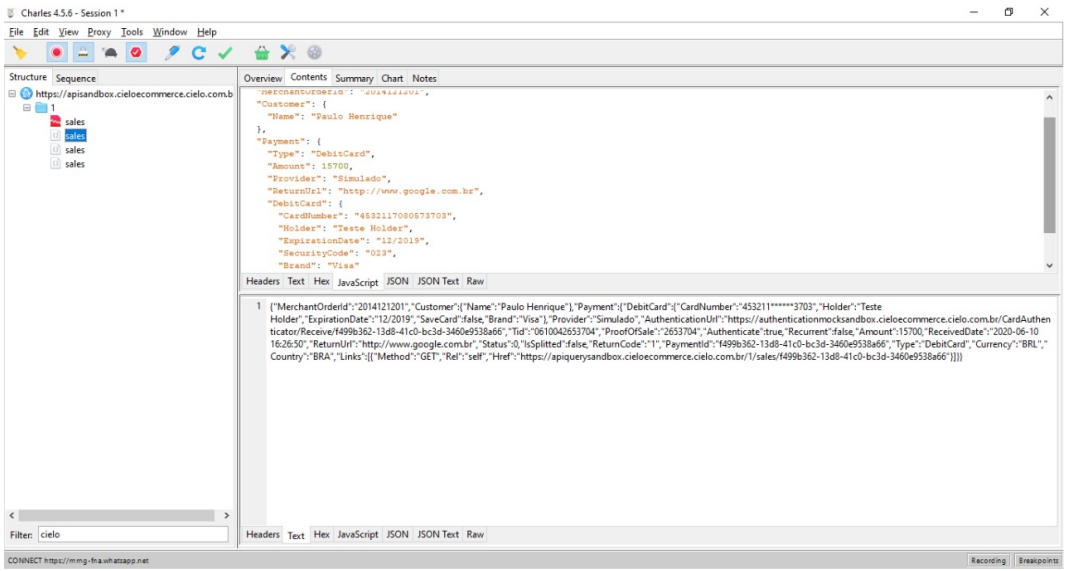


Figura 12: interceptação dos dados

5.4.2 Ataques a senhas

A segurança de senhas é um das mais importantes na área de segurança. O uso, criação e armazenamento seguro das senhas é crucial para a segurança dos usuários. Um ataque muito conhecido a senhas é o ataque de *força bruta*. Um **ataque de força bruta** ocorre quando o atacante usa técnicas para testar combinações de senhas com o objetivo de acessar a uma conta ou sistema da vítima. Esses ataques geralmente são mais bem-sucedidos nos casos em que são usadas senhas fracas ou previsíveis.

Foram testados 2 softwares capazes de fazer esse tipo de ataque, o John the ripper e o Kraken, que pode ser visto na Fig.13).

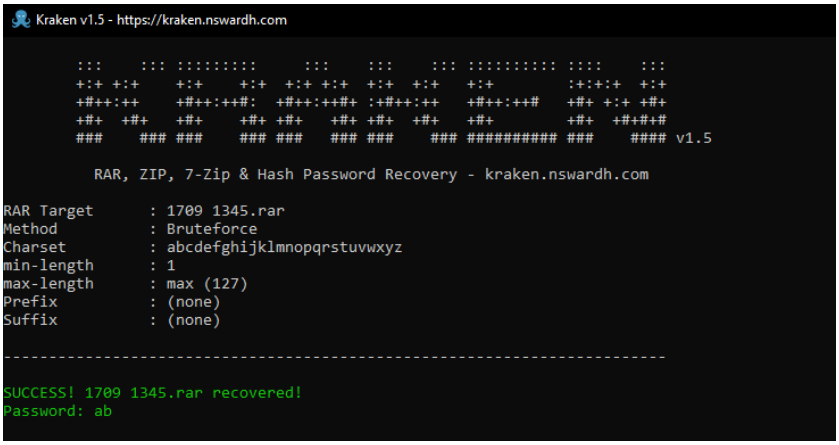


Figura 13: Utilização do Kraken para quebrar senha de arquivo compactado

5.4.3 Ataque man-in-the-middle

man-in-the-middle é um ataque em que os dados trocados são interceptados, registrados e possivelmente alterados pelo atacante sem que as vitimas notem. Durante o ataque, a comunicação é interceptada pelo atacante e retransmitida. O atacante pode decidir retransmitir entre os legítimos participantes os dados inalterados, com alterações ou bloquear partes da informação. O teste foi realizano no Software Ettercap (Fig.14).

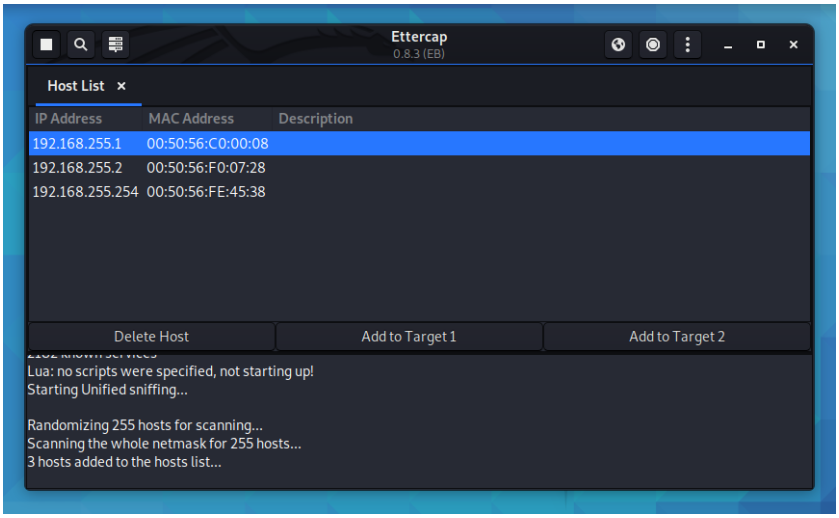


Figura 14: Ettercap, ferramenta de segurança de rede

5.4.4 Outros ataques

Além dos ataques citados acima, foram estudados muitos outros tipos de ataques conhecidos, como:

- Eavesdropping - É um tipo de ataque de interceptação de dados por meio de ferramentas e vulnerabilidades em comunicações de usuários. um exemplo é o man-in-the-middle já citado.
- Phishing - Através da engenharia social o atacante envia e-mails falsos para conseguir informações ou direcionar a websites falsos.

- Denial-of-Service (DoS) - é uma tentativa de tornar um recurso de um sistema indisponível.
- Distributed Denial of Service (DDoS)- é uma tentativa de tornar um recurso de um sistema indisponível usando vários.
- SQL Injection - é um ataque que explora a interação com base de dados através de SQL.
- Ataques a falhas de seguranças em sites e aplicativos famosos (Telegram, Facebook, Outlook, etc).

5.5 Segurança em dispositivos de Pagamento Móveis

O pagamento móvel tornou-se uma tendência geral no mundo e constitui uma parcela cada vez mais substancial dos pagamentos.Porém, também traz muitos problemas de segurança consigo, que podem causar grandes perdas, golpes, etc. Nesse contexto, Foram pesquisados alguns métodos de segurança utilizados em dispositivos de pagamento online [LWP20], com o objetivo de informar e mostrar sua importância.

- **Tokenização** - A tokenização é um processo de substituição de um PAN tradicional do cartão bancário por um token com um valor numérico exclusivo. O token de pagamento pode ser usado em todos os aspectos das transações com cartão bancário e em todos os setores, para ter versatilidade. O procedimento de geração de token é realizado pelo provedor de serviços de token (TSP). Quando o TSP está emitindo um token, ele pode flexibilizar algumas restrições, como o uso para empresas individuais, aplicativo online, aplicativo offline etc. Ele também pode limitar o valor, o tempo e o local dos tokens. Quando necessário, os tokens podem ser destruídos e reemitidos.
- **A segurança da ligação ao PAN** - A ligação PAN é o processo pelo qual o usuário vincula o cartão bancário para pagamento móvel. A segurança da ligação ao PAN é proteger as informações confidenciais do usuário no processo de ligação.
- Autenticação segura de pagamento - O pagamento remoto é um método pelo qual o terminal móvel do pagador pode concluir o pagamento longe do beneficiário. Essa abordagem é semelhante ao pagamento da página da web. Existem dois tipos de métodos de autenticação para pagamento remoto, um é autenticação de código PIN (Número de identificação pessoal) e o outro é identificação biométrica. A maior diferença entre esses dois tipos é que o primeiro precisa autenticar o PIN no servidor remoto, enquanto o segundo geralmente é autenticado localmente.
- **HTTPS** - HTTPS é uma implementação do protocolo HTTP sobre uma camada adicional de segurança que utiliza o protocolo SSL/TLS.
- **SSL/TLS** - O SSL (Secure Sockets Layer) e seu sucessor TLS (Transport Layer Security) são protocolos de criptografia projetados para internet. Permitem a comunicação segura entre os lados cliente e servidor de uma aplicação. Uma vez que o cliente e o servidor tenham decidido usar TLS/SSL, eles negociam um estado de conexão usando um procedimento de handshaking, no qual o cliente e o servidor concordam em vários parâmetros utilizados para estabelecer a conexão segura.

6 Conclusões

Nesse primeiro ano, o bolsista adquiriu conhecimentos e práticas de desenvolvimento e segurança em dispositivos móveis, estimulou a atenção em segurança no desenvolvimento de aplicativos de outros alunos da UENF. Como parte da implementação da pesquisa, fora desenvolvidos alguns pequenos aplicativos para familiarizar-se com alguns conceitos de segurança.

As atividades desenvolvidas, mesmo com as limitações da pandemia COVID-19, estimulou a pesquisa em novas área da segurança para dispositivos móveis, para as quais é necessário entender novos aspectos e implementar novas técnicas de segurança em dispositivos Android. Entre estas atividades consideramos importante abordar técnicas de Testes de Invasão ou penetração (Pentest ou segurança ofensiva), para o qual solicitamos a renovaçaõ da bolsa de Iniciação científica.

7 Perspectivas de continuidade

Como uma proposta para continuação do projeto, sugere-se aprimorar as práticas e tecnicas de segurança através do estudo em segurança ofensiva, isso é, testes de penetração (Pentest). *Pentest* é uma forma de estudar as vulnerabilidades dos sistemas e redes, afim de, posteriormente criar métodos de prevenção e correção dos mesmos.

8 Participação em Eventos

Foram desenvolvidos sete aplicativos não publicados citados na subseção 5.2

Também foi submetido e aprovado um resumo sobre *"Segurança em dispositivos de pagamento móvel"* no XII CONFICT. No trabalho são apresentados quatro métodos utilizados em dispositivos de pagamento móvel.

Referências

[AA19] S. J. Alsunaidi and A. M. Almuhaideb. The security risks associated with imeis and security solutions. In *2019 2nd International Conference on Computer Applications Information Security (ICCAIS)*, pages 1–5, May 2019. Citado na página 1.

[ALRR17] Milad Taleby Ahvanooey, Qianmu Li, Mahdi Rabbani, and Ahmed Raza Rajput. A survey on smartphones security: Software vulnerabilities, malware, and attacks. *IJACSA - International Journal of Advanced Computer Science and Applications*, 8(10):30–45, 01 2017. Citado na página 1.

[CGI12] CGI-Comitê Gestor da Internet no Brasil. *Cartilha de Segurançaa para a Internet*. CERT.br - Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil, São Paulo, SP, 2 edition, 2012. Citado na página 8.

[DDW16] Paul Deitel, Harvey Deitel, and Alexander Wald. *Android 6 for programmers : an app-driven approach*. Prentice Hall Deitel, Boston, MA, 3 edition, 2016. Citado na página 2.

[LWP20] W. Liu, X. Wang, and W. Peng. State of the art: Secure mobile payment. *IEEE Access*, 8, January 2020. Citado na página 11.

[Mor16] Avelino Morganti. Whaves: Plataforma hardware-software para internet das coisas. December 2016. Citado na página 7.

[Mor17] Daniel Moreno. *Pentest em Redes Sem Fio*. Novatec, São Paulo, SP, 1 edition, 2017. Citado na página 8.

[She20] Heidi Shey. The State of Data Security and Privacy,2020 . techreport, Forrester, Cambridge, MA, February 2020. Citado na página 1.

[SRS20] Clément Saad, Vivien Raoul, and Stéphane Saad. Mobile Application Security Guide - From development to operations. techreport, Pradeo, San Francisco, CA, 2020. Citado na página 1.

[Wei14] Georgia Weidman. *Testes de Invasão: Uma introdução ao hacking*. Novatec, São Paulo, SP, 1 edition, 2014. Citado na página 2.

9 Conforme

Campos dos Goytacazes, 30 de Setembro de 2020

José Lucio Azevedo

Aluno Bolsista

Professor Orientador