FACULDADE DE TECNOLOGIA DE AMERICANA – MINISTRO RALPH BIASI

Curso Superior de Tecnologia em Segurança da Informação

Rafael Sanches Rocha

FERRAMENTA DE SIMULAÇÃO DE INTERFACE DE LINHA DE COMANDO PARA APRENDIZADO DE SEGURANÇA OFENSIVA

Americana, SP

2020

FACULDADE DE TECNOLOGIA DE AMERICANA – MINISTRO RALPH BIASI

Curso Superior de Tecnologia em Segurança da Informação

Rafael Sanches Rocha

FERRAMENTA DE SIMULAÇÃO DE INTERFACE DE LINHA DE COMANDO PARA APRENDIZADO DE SEGURANÇA OFENSIVA

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Segurança da Informação, sob a orientação da Prof.(a) Dra. Maria Cristina Aranda.

Área de concentração: Segurança da Informação.

Americana, SP

2020

FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS
Dados Internacionais de Catalogação-na-fonte

R576f ROCHA, Rafael Sanches

Ferramenta de simulação de interface de linha de comando para aprendizado de segurança ofensiva. / Rafael Sanches Rocha. – Americana, 2020.

62f.

Monografia (Curso Superior de Tecnologia em Segurança da Informação) - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza

Orientador: Profa. Dra. Maria Cristina Aranda

1 Segurança em sistemas de informação 2. Aprendizado de máquina I. ARANDA, Maria Cristina II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana

CDU: 681.518.3

Rafael Sanches Rocha

FERRAMENTA DE SIMULAÇÃO DE INTERFACE DE LINHA DE COMANDO PARA APRENDIZADO DE SEGURANÇA OFENSIVA

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Segurança da Informação pelo CEETEPS/Faculdade de Tecnologia – FATEC/ Americana.
Área de concentração: Segurança da Informação.

Americana, 30 de Junho de 2020.

Banca Examinadora:

_____
Maria Cristina Aranda (Presidente)
Doutora
FATEC Americana

_____
José Luís Zem (Membro)
Doutor
FATEC Americana

_____
Rogério Nunes de Freitas (Membro)
Mestre
FATEC Americana

AGRADECIMENTOS

Agradeço a Deus, pelos desafios com os quais me fortaleceu, pelas realizações que me concedeu e pelos novos planos que me trouxe.

Agradeço aos meus pais, César e Denise, pelo apoio irrestrito para que eu pudesse trilhar o caminho da educação. À minha mãe, por todo carinho e cuidado materno, e ao meu pai, por toda a orientação e apoio fornecidos. Sem vocês, nada seria.

Agradeço à minha namorada, Jainara Silva Rodrigues, pela fonte constante de companheirismo. Presente em grande parte desse caminho, sempre esteve encorajando meus planos.

Agradeço a toda experiência compartilhada junto aos colegas da graduação. Diversas foram as passagens em grupo, mas em especial agradeço ao Guilherme Fontes Pereira, em uma etapa mais inicial do curso, e ao Dener Aparecido Caldeira Paschoal, em uma etapa mais próxima à conclusão do curso, por terem sido os colegas com quem mais vivenciei essa jornada.

Agradeço a todos os professores que fizeram parte dessa trajetória. Muitos foram os que marcaram positivamente, seja pelo empenho docente ou pela boa relação estabelecida. Em especial, agradeço ao professor Benedito Luciano Antunes de França – Benê – pelas incontáveis horas de atendimento, ouvindo minhas dúvidas e questionamentos quanto à atuação docente, e me estimulando a seguir esse caminho. Encontrá-lo no momento certo foi fundamental para eu descobrir o que gostaria de fazer pelo resto da vida e dar os primeiros passos nessa direção. Ainda, de forma especial, agradeço à Maria Cristina Aranda, que diante de uma situação delicada na graduação se dispôs a me ajudar, e mais do que isso, a me orientar nesse trabalho. Sou grato pelo seu exemplo de apoio, de docência e empatia com os alunos.

Agradeço à FATEC Americana pela oferta desse curso, o que permitiu uma valiosa expansão de conhecimento e a concretização de mais um objetivo acadêmico.

“*A melhor defesa é um bom ataque.*”

(Autor desconhecido)

RESUMO

Este trabalho apresenta a proposição de uma ferramenta de simulação computacional de tarefas de segurança ofensiva. Segurança ofensiva representa a atuação profissional em busca de vulnerabilidades a serem exploradas em uma infraestrutura de comunicação, a fim de corrigi-las. Essas atividades podem ser complexas, inacessíveis ou apresentar riscos, principalmente para iniciantes na área de segurança da informação. Para contornar obstáculos da realidade, como limitações de recursos e riscos associados, pode-se empregar técnicas de simulação computacional a fim de abstrair elementos da realidade para representa-la em um programa com algumas limitações, diminuindo as restrições indesejadas, evidenciando o comportamento em estudo e tornando o cenário simulado mais acessível ao usuário. Com esse intuito, esse trabalho especifica, desenvolve e apresenta um simulador acessível para o aprendizado e prática de segurança ofensiva, por meio da representação da execução de fluxo de comandos em uma interface gráfica similar à de um terminal do Kali Linux. Esse fluxo representa a condução dos passos para se executar uma tarefa de segurança ofensiva. Para comprovar a eficácia da proposta, é mapeado um cenário real de quebra de senha de rede Wi-Fi, sendo incluído no simulador e demonstrando a funcionalidade desta ferramenta e sua capacidade de reprodução de cenários de teste de segurança. O simulador desenvolvido atinge os objetivos propostos, e apresenta potencial de evolução e recursos para auxiliar em tarefas didáticas de aprendizado prático de segurança da informação. Um questionário foi conduzido com alunos do curso de Segurança da Informação da FATEC Americana e foi observado alinhamento entre o que a ferramenta propõe e as expectativas e necessidades dos alunos participantes.

Palavras Chave: segurança ofensiva; simulação, segurança da informação.

ABSTRACT

This work presents the proposition of a computer simulation tool for offensive security tasks. Offensive security represents professional action in search of vulnerabilities to be exploited in a communication infrastructure, in order to correct them. These activities can be complex, inaccessible or present risks, especially for beginners in the area of information security. To circumvent reality obstacles, such as resource limitations and associated risks, computer simulation techniques can be employed in order to abstract elements of reality to represent it in a program with some limitations, reducing unwanted restrictions, focusing in the behavior under study and making the simulated scenario more accessible to the user. To this end, this work specifies, develops and presents an accessible simulator for offensive security learning and practice, by representing the execution of command flow in a graphical interface similar to that of a Kali Linux terminal. This flow represents the conduction of steps to perform an offensive security task. To prove the effectiveness of the proposal, a real Wi-Fi password cracking scenario is mapped, being included in the simulator and demonstrating the functionality of this tool and its ability to reproduce security test scenarios. The developed simulator achieves the proposed objectives, and presents potential for evolution and resources to assist in didactic tasks of practical information security learning. A questionnaire was conducted with students from the FATEC American Information Security course and an alignment was observed between what the tool proposes and the expectations and needs of the participating students.

Keywords: *offensive security; simulation; information security.*

SUMÁRIO

1. INTRODUÇÃO 14
2. REFERENCIAL TEÓRICO 17
2.1 SEGURANÇA OFENSIVA 17
2.2 SIMULAÇÃO COMPUTACIONAL 18
2.3 FERRAMENTAS EXISTENTES 19
2.4 CONSIDERAÇÕES DO CAPÍTULO 23
3. DESENVOLVIMENTO DO SIMULADOR 24
3.1 CONSIDERAÇÕES DO CAPÍTULO 31
4. ESTUDO DE CASO – ATAQUE À REDE SEM FIO 32
4.1 CENÁRIO REAL 33
4.2 CENÁRIO SIMULADO 42
4.3 CONSIDERAÇÕES DO CAPÍTULO 47
5. ANÁLISE DO QUESTIONÁRIO 48
6. CONSIDERAÇÕES FINAIS 52
REFERÊNCIAS BIBLIOGRÁFICAS 54
APÊNDICE A – QUESTIONÁRIO 56
APÊNDICE B – RESPOSTAS DO QUESTIONÁRIO 58

LISTA DE FIGURAS

Figura 1 – Cenário no Cisco Packet Tracer....20
Figura 2 – Cenário no GNS3....21
Figura 3 – Tela do jogo Hacker Wars....22
Figura 4 – Tela do site Codecademy: Learn Python....23
Figura 5 – Fluxograma do Simulador....25
Figura 6 – Menu de seleção de cenários....26
Figura 7 – Modo Aprender....26
Figura 8 – Modo Praticar....27
Figura 9 – Modo Time Attack....27
Figura 10 – Comando help....28
Figura 11 – Criação do cenário....29
Figura 12 – Arquivo de salvamento dos cenários....30
Figura 13 – Algumas características visuais do simulador....31
Figura 14 – Aircrack-ng: logotipo....32
Figura 15 – Configuração da rede Wi-Fi....34
Figura 16 – Cenário preparado....34
Figura 17 – Identificando todas as interfaces de rede....35
Figura 18 – Consultando o adaptador de rede USB....36
Figura 19 – Ativando o modo monitor....36
Figura 20 – Encerrando processos conflitantes....37
Figura 21 – Conferindo o modo da interface....37
Figura 22 – Airodump executando monitoramento....38
Figura 23 – Airodump executando em uma rede específica....39
Figura 24 – Arquivos gerados pelo airodump....39
Figura 25 – Aireplay executando deauth....40
Figura 26 – Handshake capturado....40
Figura 27 – Dicionário utilizado....41
Figura 28 – Aircrack identificando a senha....41
Figura 29 – Interface retornando ao modo normal....42
Figura 30 – Criação do cenário da simulação Wi-Fi Cracking....43
Figura 31 – Wi-Fi Cracking: Passo 1....44
Figura 32 – Wi-Fi Cracking: Passo 2....44

Figura 33 – Wi-Fi Cracking: Passo 3 45
Figura 34 – Wi-Fi Cracking: Passo 4 45
Figura 35 – Wi-Fi Cracking: Passo 5 46
Figura 36 – Wi-Fi Cracking: Passo 6 46
Figura 37 – QR Code da apresentação 53

LISTA DE GRÁFICOS

Gráfico 1 - Situação dos participantes perante o curso de SI 48
Gráfico 2 - Conhecimento dos participantes sobre segurança ofensiva.....49

LISTA DE ABREVIATURAS E SIGLAS

AP	*Access Point*
API	*Application Programming Interface*
CCNA	*Cisco Certified Network Associate*
CLI	*Command Line Interface*
GNS3	*Graphical Network Simulator-3*
GUI	*Graphical User Interface*
IDE	*Integrated Development Environment*
IOS	*Internetworking Operating System*
SED	Simulação de Eventos Discretos
SI	Segurança da Informação
TI	Tecnologia da Informação
WPA	*Wi-Fi Protected Access*

1. INTRODUÇÃO

O conhecimento e prática da segurança ofensiva é um passo fundamental para o desenvolvimento da capacidade de um profissional de Tecnologia da Informação (TI) em proteger os sistemas computacionais sob sua responsabilidade.

Uma frase de conhecimento popular, adaptada para diversas áreas, afirma que “a melhor defesa é um bom ataque”. Esse conceito é reforçado por profissionais de Segurança da Informação (SI). Segundo a Offensive Security (2020a), aprender como as vulnerabilidades de segurança são exploradas capacita o profissional a defender melhor seus sistemas. Sanabria (2018) aponta que os times de segurança, ao tentarem encontrar brechas na própria rede, podem identificar e remediar falhas na infraestrutura antes que agentes maliciosos possam explorá-las.

Contudo, tais atividades devem ser geridas com ética e prudência (BASTA, 2014). Iniciantes nos conceitos de segurança da informação podem incorrer em práticas ilegais na ânsia pelo aprendizado, seja intencionalmente, em busca de maiores desafios, ou mesmo por descuido, causado pelo desconhecimento técnico e dos limites legais. O aluno ou aprendiz dessas práticas pode cometer descuidos técnicos, justamente pela falta do conhecimento que busca obter, resultando em brechas de segurança permanentes e mais críticas, que não tinha intenção em criar.

Esse retrato representa uma necessidade de um conhecimento que exige grande prática, ao mesmo tempo que demanda cautela e prudência de quem busca esse conhecimento, qualidades estas que podem ainda não estar amadurecidas no praticante. Mais do que isso, essa prática exige um ambiente que pode se tornar complexo, demandando a presença e configuração de diferentes dispositivos e sistemas, como *switches*, servidores, sistemas operacionais, bancos de dados, entre outros itens que compõem uma infraestrutura de comunicação.

Diante desse cenário, há diferentes atitudes que um iniciante pode adotar: de alguma forma adquirir esses equipamentos e montar seus próprios cenários de teste, o que é pouco provável; testar e desenvolver seus conhecimentos na infraestrutura de terceiros, atuando de forma ilegal; permanecer com cenários mais básicos que estejam ao seu alcance e das máquinas virtuais que possa criar em seu próprio computador; desmotivar e desistir da evolução na área de segurança.

Esse impasse não é exclusivo da área de segurança da informação. Muitas outras atividades exigem prática, treino, repetição, sem ter ao dispor a presença constante dos recursos relacionados ou tendo que evitar riscos associados. Um piloto amador pode praticar horas de voo sem ter ao seu dispor um Boeing. Um astronauta pode treinar e se preparar exaustivamente antes de ir para órbita. Um candidato a carteira de motorista consegue praticar antes de entrar em um carro. E essas atividades possuem riscos relacionados a sua má execução. Todos esses exemplos podem encontrar uma alternativa na mesma abordagem: simulação computacional.

A modelagem matemática e simulação computacional visam abstrair características da realidade e inseri-las em uma simulação (BRITANNICA, 2020). Dessa forma, apresentam a possibilidade de reproduzir os efeitos reais desejados, sem os efeitos indesejados. Um piloto pode praticar todas as operações de uma cabine de avião, sem o risco de derrubá-lo. Um astronauta pode praticar todos os procedimentos necessários na nave, sem o risco de ficar à deriva no espaço. Um candidato à carteira de motorista pode aprender sem o risco de atropelar os pedestres. E um profissional de segurança da informação pode praticar ataques e invasões em infraestruturas, sem invadir sistemas alheios, deixar brechas indesejadas em sua própria estrutura, ou mesmo sem a necessidade de criação e configuração exaustiva prévia desse ambiente.

Esse trabalho foi concebido com o intuito de auxiliar nessa tarefa de prática de segurança ofensiva, ao especificar, desenvolver e oferecer uma ferramenta de simulação moldada aos praticantes de segurança ofensiva. Um simulador que permita a prática similar à realidade, sem a necessidade de aquisição de equipamentos, trabalho para configuração do ambiente e preocupações com os riscos associados.

Dessa forma, os objetivos desse trabalho são:

- Desenvolver uma ferramenta que simule uma interface de linha de comando, que é a aplicação pela qual as atividades de segurança são executadas.
- Selecionar e mapear o processo de execução de uma tarefa de segurança ofensiva real, adaptando-a para a simulação, comprovando assim a sua funcionalidade.
- Aplicar um questionário com o público-alvo dessa ferramenta (i.e., os alunos de SI da FATEC) para colher impressões e recomendações sobre a proposta dessa ferramenta.

A motivação desse trabalho foi a possibilidade de deixar uma contribuição ao curso que tanto me ofereceu em aprendizado. Identificar uma lacuna em uma área pela qual tenho tanto apreço, e realizar uma primeira contribuição ainda que modesta, é uma realização acadêmica e pessoal.

A estrutura desse texto foi organizada na seguinte forma: O Capítulo 2 apresenta o Referencial Teórico, introduzindo a Segurança Ofensiva e a Simulação Computacional. Algumas ferramentas existentes são apresentadas para exemplificar o potencial da simulação. O Capítulo 3 discorre sobre o desenvolvimento do simulador, apresentando características da ferramenta introduzida por esse trabalho. O Capítulo 4 apresenta a execução de um cenário real de prática de segurança ofensiva, e a sua respectiva simulação executada na ferramenta desenvolvida. O Capítulo 5 analisa os dados obtidos pelo questionário aplicado, traçando comparações com a simulação realizada. O Capítulo 6 apresenta as conclusões do trabalho e sugestões para trabalhos futuros.

2. REFERENCIAL TEÓRICO

As bases de referência teórica para este trabalho residem, primeiramente, na compreensão da Segurança Ofensiva como área da Segurança da Informação. Em seguida, são apresentados conceitos de Simulação Computacional, como meio para se alcançar o objetivo deste trabalho. Por fim, são exemplificadas algumas ferramentas baseadas em simulação para práticas relacionadas à segurança da informação.

2.1 SEGURANÇA OFENSIVA

Segurança Ofensiva é um termo surgido para contrastar com a tradicional “segurança defensiva” adotada por muitas equipes de TI. Em uma abordagem defensiva, as equipes se pautam em medidas preventivas como manter sistemas atualizados, instalar e configurar antivírus e *firewalls*, e monitorar o tráfego da rede. Contudo, nessa abordagem, a equipe de TI permanece aguardando por um evento malicioso, esperando que suas políticas e procedimentos estejam adequados para conseguir reagir no momento do ataque.

Em contrapartida, com uma abordagem de segurança ofensiva, a equipe constantemente busca brechas em seus sistemas, atuando de forma semelhante ao invasor, partindo do lado externo da rede. Ao tentar atacar sua própria infraestrutura, com as mesmas ferramentas e a mentalidade do agente malicioso, pode encontrar falhas que não seriam facilmente identificadas de uma perspectiva interna.

O termo Segurança Ofensiva é presente em cursos (UDEMY, 2020), em conferências (OFFENSIVE CON, 2020), em serviços de grandes empresas (IBM, 2020), e provavelmente tenha sido popularizado pela empresa que carrega esse nome, a Offensive Security (OFFENSIVE SECURITY, 2020b), responsável pela criação e distribuição de Linux direcionada para práticas de segurança ofensiva, o Kali Linux (OFFENSIVE SECURITY, 2020c). Baseado em Debian, foi lançado em 2013, substituindo o projeto anterior – Backtrack – que havia sido lançado em 2006 e descontinuado em 2012. Além disso, a Offensive Security promove uma das certificações em segurança ofensiva mais respeitadas do mercado, a Offensive Security Certified Professional (OSCP) (OFFENSIVE SECURITY, 2020d).

Segurança Ofensiva é um nome que possui alguns sinônimos e conceitos relacionados. Talvez os termos mais conhecidos sejam Teste de Vulnerabilidade, Teste de Intrusão ou Teste de Penetração (do inglês *Penetration Test*, ou ainda, *Pentest*), que consiste na ação de buscar vulnerabilidades em uma rede e emitir um relatório de diagnóstico de sua segurança, apontando as falhas, suas consequências se exploradas, e as correções a serem adotadas (HACKER SECURITY, 2020).

Outro conceito existente é *Red Team* para se referir a uma equipe que possui autorização para invadir um sistema ou rede, enquanto o *Blue Team* denomina a equipe que permanece defendendo a infraestrutura (CYBRARY, 2015). Essa caracterização é utilizada em *wargames* e testes de segurança das redes de comunicação.

Por fim, os termos *White Hat Hacker* ou *Ethical Hacker* classificam a postura de um profissional de segurança da informação com *know-how* de como explorar falhas de segurança e invadir sistemas (EC-COUNCIL, 2020). Porém, ao invés de buscar benefícios próprios com tais invasões, este utiliza seus conhecimentos técnicos para fins legais, seja contratado ou exercendo consultoria para empresas, realizando *pentesting*, ou mesmo deliberadamente buscando falhas em sistemas e sites e os reportando aos responsáveis, para que as correções sejam feitas.

Explorando esses termos, muitas referências surgem, de fóruns e comunidades dessa temática, além de livros, tutoriais, cursos, certificações, e diversos materiais que contribuem para o aprofundamento do profissional de TI na área de segurança ofensiva.

2.2 SIMULAÇÃO COMPUTACIONAL

Segundo Banks *et al*. (2005) simulação é a imitação da operação de processos do mundo real ou sistemas ao longo do tempo, seja feita em computador ou não. Quando conduzida computacionalmente, essa ação pode ser utilizada para inferir características do sistema real, a partir de dados obtidos pela simulação, ou para representar em uma simulação eventos abstraídos da realidade. Dessa forma, para este trabalho, utilizou-se a simulação computacional para representar uma interface de linha de comando, permitindo ao usuário digitar comandos e visualizar resultados, de forma semelhante à interação com um terminal real.

Simulações são técnicas poderosas que podem ser utilizadas em diferentes pesquisas, muitas vezes, inviáveis de serem conduzidos na realidade. O estudo de previsão meteorológica (INMET, 2020) e de fluxo de pedestres em estações de metrô (ENGIMIND, 2017) são apenas alguns exemplos de aplicações de simulações computacionais.

Abordagens para representações de fenômenos mais complexos se pautam na modelagem matemática para caracterizar numericamente em sistemas computacionais esses eventos do mundo real. A Simulação de Eventos Discretos (SED) consiste na especificação de um sistema no qual são modeladas variáveis da realidade que se deseja representar, simulando condições a serem testadas computacionalmente e oferecendo um meio para analisar soluções de difícil implantação e teste prático.

Segundo Law e Kelton (1999), simulações de eventos discretos são técnicas computacionais para imitar operações ou processos reais de todos os tipos, chamados de sistemas. É necessário fazer suposições sobre como funciona, por meio de relações lógicas ou matemáticas, que correspondem ao modelo gerado para se estudar e obter alguma compreensão de como o sistema se comporta. Muitos sistemas do mundo real são complexos e devem ser estudados por simulação, obtendo dados numéricos para estimar as características reais do sistema.

Portanto, a criação de uma simulação é o meio viável para a solução proposta por esse trabalho, porém, nesse estudo destaca-se que não será abordado um processo estocástico ou a representação de variáveis, dessa forma, não é pretendido a obtenção de nenhum valor para inferir ao sistema real. O intuito desse trabalho é adotar uma abordagem mais simples de simulação computacional, suficiente para a representação da entrada, processamento e saída dos comandos no terminal, a fim de observar na simulação parte do comportamento do sistema real.

2.3 FERRAMENTAS EXISTENTES

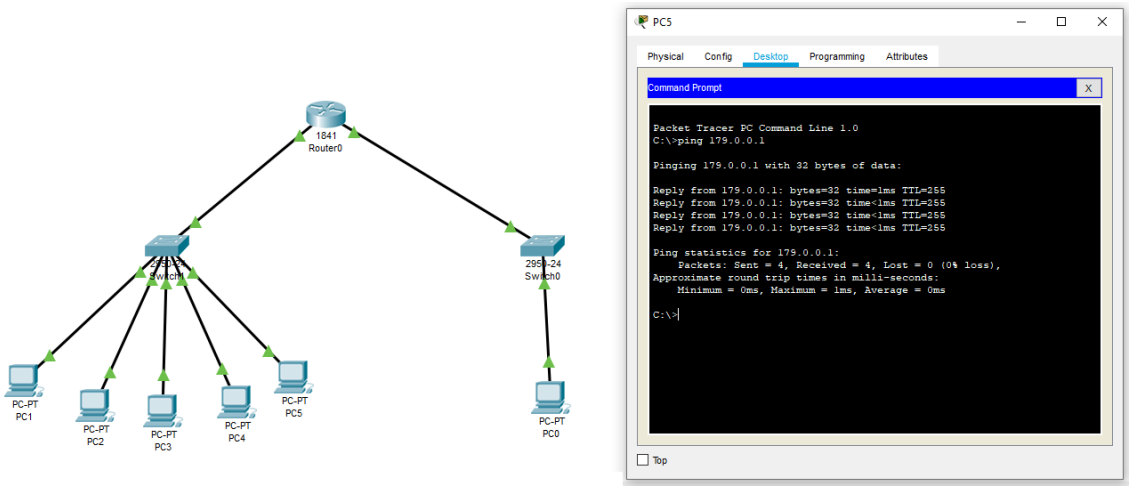
Buscando por ferramentas de simulação relacionadas a atividades de segurança, não foi possível encontrar alguma específica com essa finalidade, porém algumas ferramentas relacionadas foram identificadas.

O Cisco Packet Tracer (CISCO, 2020a), atualmente na versão 7.3.0, é um simulador de redes de computadores, que permite recriar topologias com *switches*,

roteadores, servidores, *desktops*, pontos de acesso à rede sem fio, além de configurar protocolos de redes e os vários equipamentos disponíveis. Essa é uma ferramenta muito utilizada para aprendizado, sendo a referência de prática para a certificação Cisco Certified Network Associate (CCNA).

A Figura 1 representa uma topologia criada no Packet Tracer. É possível criar uma interação com um *desktop* (nesse caso, o PC5), abrindo um *prompt* de comando para digitar linhas de comando. Dessa forma, a interação é semelhante à proposta nesse trabalho, porém voltada à configuração e diagnóstico de redes.

Figura 1 – Cenário no Cisco Packet Tracer



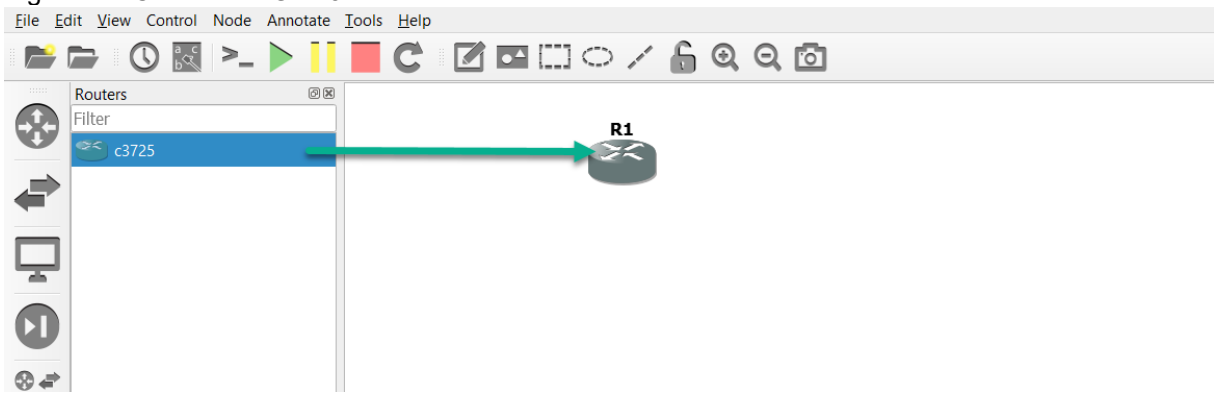
Fonte: Elaborado pelo autor.

Para estudos mais aprofundados, como a preparação para certificações mais avançadas, outras ferramentas são indicadas, como o Graphical Network Simulator-3, ou GNS3 (GNS3, 2020), atualmente na versão 2.2.9. Mais do que simular alguns comportamentos de rede, esse *software* permite emular os sistemas dos dispositivos. Por exemplo, o Cisco Internetworking Operating System (IOS) (CISCO, 2020b) é o sistema operacional de muitos roteadores da Cisco, e o GNS3 permite que seja carregada uma imagem desse sistema operacional, para ser emulada na topologia.

Nesse ponto, distingue-se simulação de emulação. Uma simulação é a imitação de um sistema, podendo representar apenas algumas partes, e sendo implementando de forma distinta da implantação do sistema real. Uma emulação é a reprodução do sistema real, onde é esperado que se comporte exatamente e inteiramente como o sistema representado. Por isso, na emulação de sistemas operacionais, utiliza-se o carregamento da imagem original presente nos dispositivos físicos.

A Figura 2 representa a criação de um roteador no GNS3. É carregada uma imagem IOS c3725 no roteador R1. Há uma grande variedade de imagens de sistemas que podem ser encontradas na Internet para serem baixadas e carregadas no GNS3, para emular cada dispositivo desejado.

Figura 2 – Cenário no GNS3

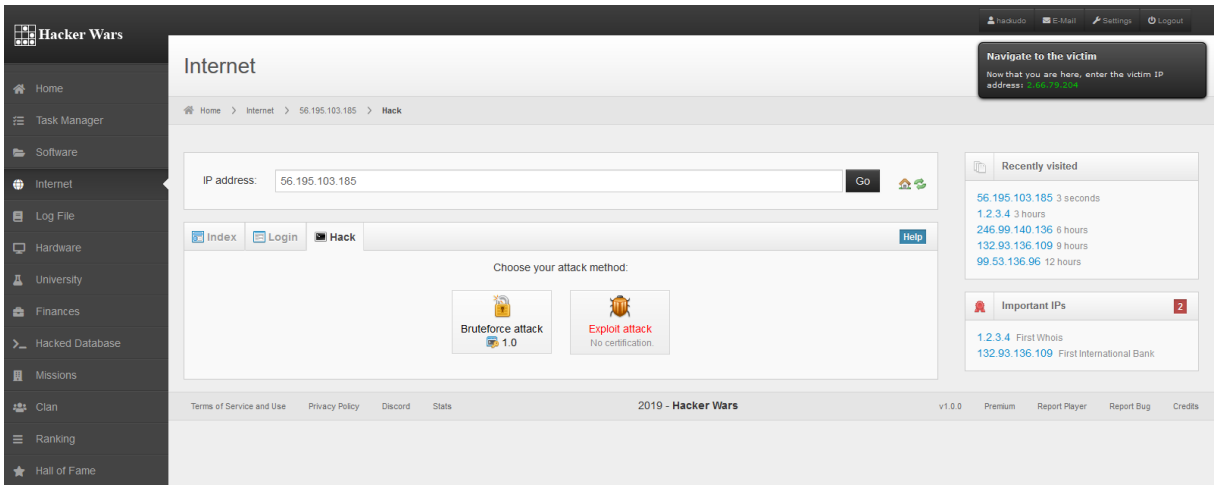


Fonte: GNS3, 2019.

Mais próximo de tarefas de segurança ofensiva, ou mais precisamente com a temática *Hacking*, foram encontradas simulações com diferentes graus de abstração, voltadas para o entretenimento, ou seja, jogos digitais. Há uma grande variedade de títulos, desde jogos altamente populares e com alto grau de abstração da realidade, como *Watch Dogs*, à títulos *indie* que se aproximam um pouco mais da realidade (ainda que de forma bastante abstrata).

Dentre esses títulos, há o jogo brasileiro Hacker Experience. Inicialmente publicado em 2014 por um único desenvolvedor, o intuito do jogo é representar tarefas de *hacking* mas com o objetivo de entreter, sem necessitar de conhecimentos técnicos e nem representar conexão com a realidade. O jogo funciona a partir de um navegador e utiliza menus e funcionalidades simples para interação do jogador, representando a execução de tarefas complexas. A Figura 3 ilustra uma técnica representada de forma simples: para um ataque de *brute force*, basta digitar o IP da vítima, e clicar em um botão para executar o ataque e automaticamente obter acesso. Portanto, apesar de representar uma simulação do que poderia ser a tela do computador de um *hacker*, o intuito do jogo é entreter, apresentando termos da área de segurança ofensiva, mas sem correlação aos passos da execução real.

Figura 3 – Tela do jogo Hacker Wars



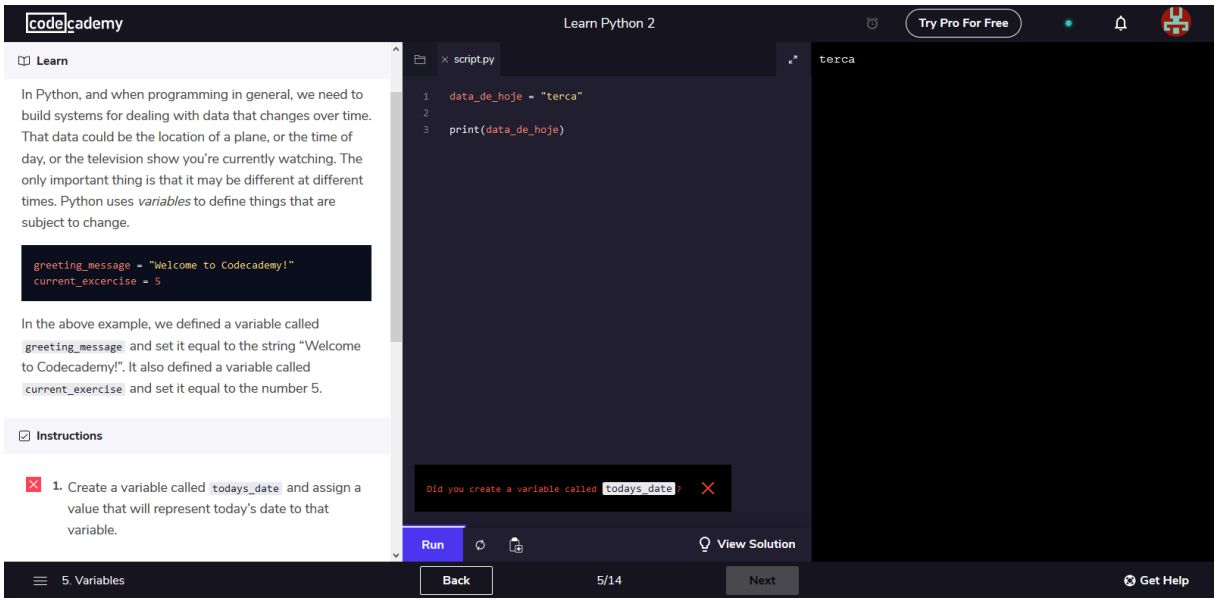
Fonte: Elaborado pelo autor.

O jogo Hacker Experience foi descontinuado pelo desenvolvedor original e o código-fonte foi liberado publicamente. Um *fork* do projeto foi realizado, algumas atualizações efetuadas, e hoje o jogo encontra-se pelo nome de Hacker Wars (HACKER WARS, 2019).

Por fim, no campo lúdico, entre diversos *sites* destinados ao aprendizado de programação e conceitos de TI, pode-se mencionar o Codecademy (CODECADEMY, 2020). Essa é uma plataforma destinada ao aprendizado e prática de linguagens de programação. Para isso, oferece um painel semelhante à uma interface de linha de comando, sendo o meio pelo qual o usuário pode digitar e praticar os comandos aprendidos.

Há uma trilha de aprendizado para Bash Scripting, a sintaxe adotada no uso do terminal proposto nesse trabalho, porém esse é um curso pago na plataforma. A Figura 4 ilustra uma tela do curso de Python, demonstrando que há uma área para instruções, uma para digitação dos comandos, e outra para a saída representada pela execução do código. Nessa figura observa-se que não foi atendido a instrução esperada para esse passo, por isso uma mensagem de alerta é gerada, aguardando até que o usuário digite algo válido para cumprir essa tarefa.

Figura 4 – Tela do site Codecademy: Learn Python



Fonte: Elaborado pelo autor.

Por essa investigação, não foi identificada uma ferramenta com o mesmo objetivo da proposta nesse trabalho. Portanto, esse trabalho busca simular uma interface de linha de comando semelhante como é feito pelo Packet Tracer, com a temática de segurança ofensiva abordada pelo Hacker Wars, e o foco no ensinamento de cada etapa como no Codecademy.

2.4 CONSIDERAÇÕES DO CAPÍTULO

Segurança ofensiva consiste na ação de busca por vulnerabilidades de segurança na infraestrutura de comunicação, a fim de encontrá-las e corrigi-las antes que um agente malicioso possa explorá-las. Para isso, é necessário conhecer e dominar as atividades que estes mesmos agentes executariam.

Como tais atividades são executadas por meio de uma interface de linha de comando, é interessante que, para aprendizado e prática dessas ações, uma ferramenta represente um terminal para que o aluno ou aprendiz possa praticar em um ambiente simulado. Simulações computacionais são recursos potentes e amplamente difundidos para diferentes cenários de estudo, representando uma possibilidade real, e ainda pouco explorada, de aplicação no contexto de segurança ofensiva.

3. DESENVOLVIMENTO DO SIMULADOR

Para atingir o objetivo de construção de uma ferramenta de simulação para prática de segurança ofensiva, primeiro deve ser identificado qual ferramenta ou recurso do mundo real que será simulado. A distribuição Kali Linux é amplamente reconhecida e adotada em práticas de segurança ofensiva. Dessa forma, este trabalho se propõe a simular a interface de linha de comando desse sistema operacional.

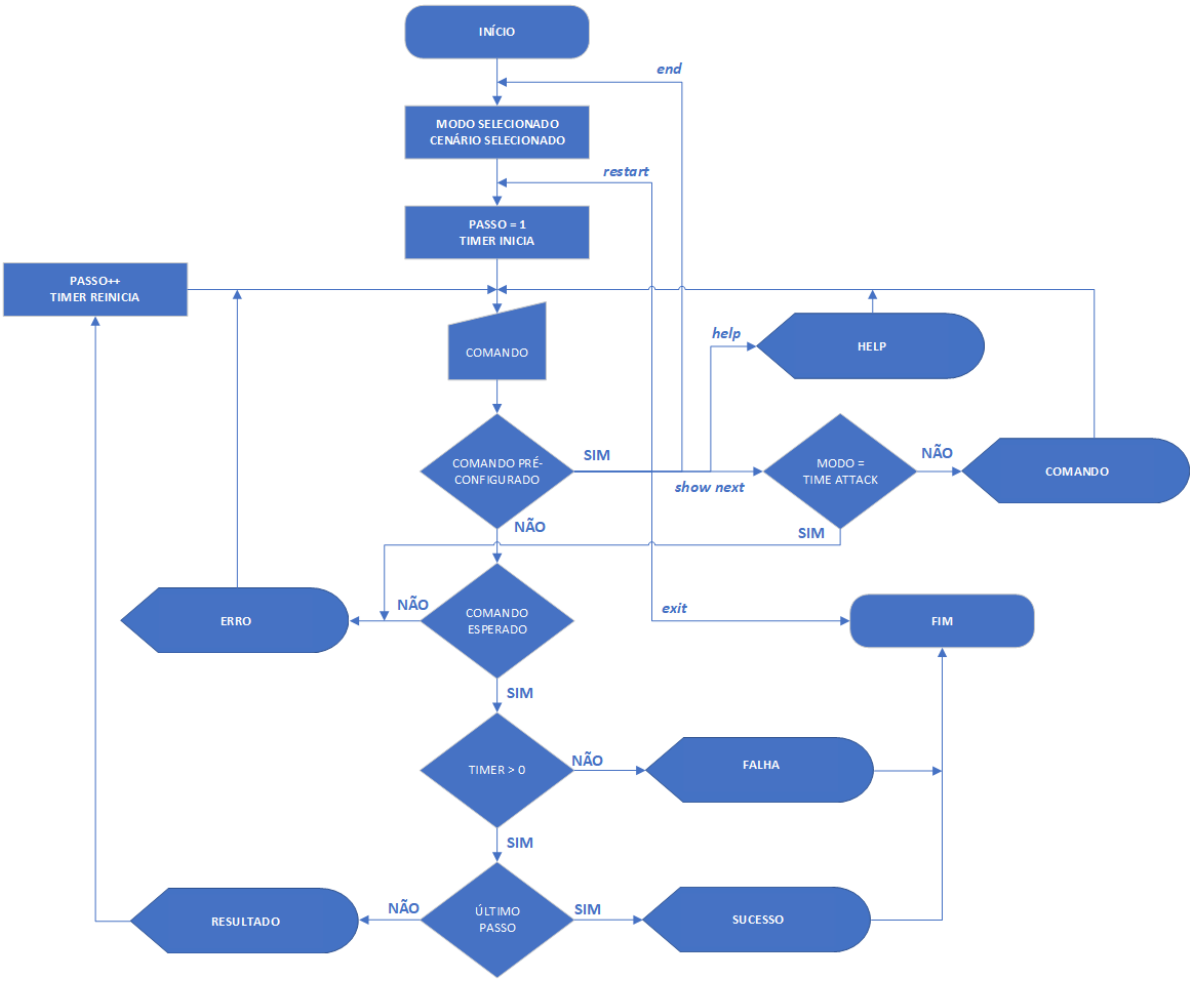
A terminologia para se referir à interface de linha de comando, ou *Command Line Interface* (CLI) é abrangente (STACK EXCHANGE, 2010). Basicamente, CLI é o nome dado ao programa que permite a entrada de comandos a serem executados pelo sistema operacional. *Shell* é o interpretador desses comandos, sendo o *Bash* comum nas distribuições do Linux. *Terminal* (ou tty) refere-se ao ambiente lógico de interação entre usuário e sistema, enquanto *Console* refere-se a um equipamento de interface de usuário, sendo por exemplo, o teclado. Nesse trabalho, CLI e Terminal são adotados como sinônimos, referindo-se ao programa com o qual o usuário pode inserir comandos, executá-los e visualizar as informações de saída.

O segundo passo é a definição do meio de desenvolvimento da ferramenta proposta. Foi escolhida a linguagem de programação Java (ORACLE, 2020) por apresentar a *Application Programming Interface* (API) SWING, que entrega acesso a um amplo conjunto de recursos para construção de interfaces gráficas, o que é uma necessidade central na construção desse simulador.

Por se tratar de uma simulação, o que se subentende ser uma representação da realidade, deve-se definir o que será representado e o nível de abstração adotado. De forma resumida, a ferramenta desenvolvida deve simular a interface gráfica do Terminal, portanto, representar um fundo preto com letras brancas, incluindo a referência do usuário e diretório em cada linha, aceitando a digitação de comandos a cada ENTER pressionado. Deve simular características do Terminal, como o sinal de *underline* piscando após o último caractere digitado, uma barra de rolagem dinâmica que aparece quando as mensagens extrapolam a dimensão da tela, e alguns comandos básicos a serem identificados em qualquer contexto de execução.

Além da simulação do Terminal, a ferramenta deve prover outros recursos inerentes à sua proposta: permitir a criação de cenários de teste, que justamente serão simulados na ferramenta. A partir de todas essas necessidades discutidas, define-se o fluxo de funcionamento do simulador representado na Figura 5.

Figura 5 – Fluxograma do Simulador



Fonte: Elaborado pelo autor.

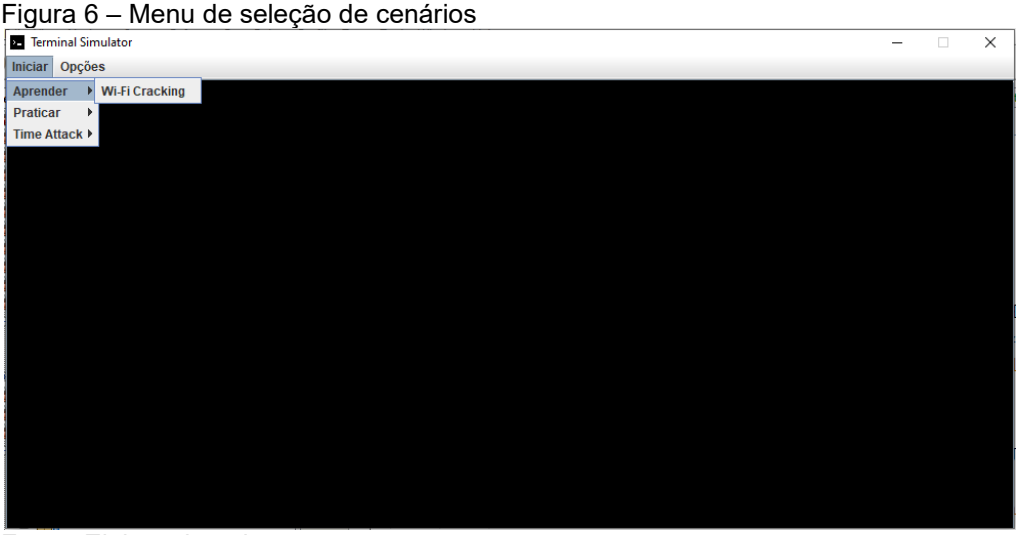
A verificação do *timer* somente será feita quando em modo “Time Attack”. Outro ponto de observação é que o simulador permanece funcional mesmo sem carregar um cenário de simulação. Nesse caso somente os comandos pré-configurados serão verificados.

O simulador possui um total de 3 modos:

1. *Aprender*: a cada passo são impressos o número do passo, comando a ser executado e explicação do comando.
2. *Praticar*: não é mostrado nenhuma mensagem a cada passo, se assemelhando ao Terminal real.
3. *Time Attack*: semelhante ao modo Praticar, com o acréscimo de um contador no canto superior direito do Terminal. Cada passo deve ser executado antes do término desse contador.

A Figura 6 ilustra o menu de seleção dos cenários. Os mesmos cenários aparecem selecionáveis para os três modos. Nesse exemplo, há apenas um cenário criado, que é apresentado na Seção 4.2. Com a criação de novos cenários, estes apareceriam listados nesse menu.

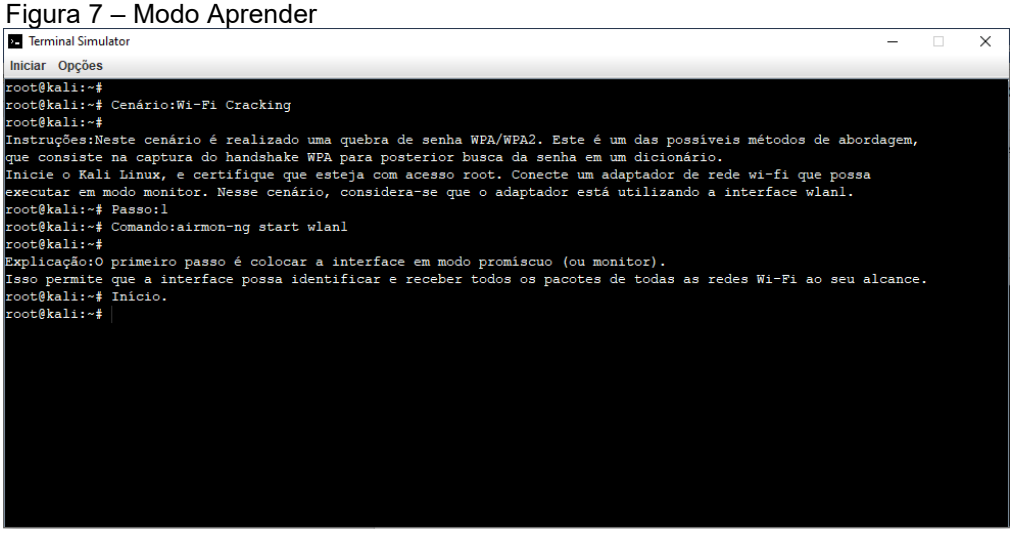
Figura 6 – Menu de seleção de cenários



Fonte: Elaborado pelo autor.

A Figura 7 ilustra o carregamento do cenário *Wi-Fi Cracking* pelo modo Aprender. Todas as informações de cada passo são carregadas, por este ser um modo para iniciantes. Nota-se que no início é impresso o nome do cenário e as instruções gerais. Esses dois itens são apresentados no início dos três modos.

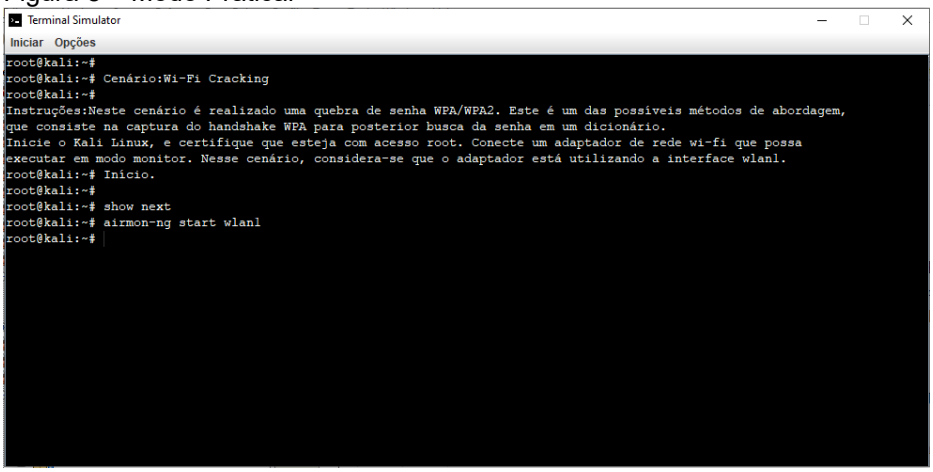
Figura 7 – Modo Aprender



Fonte: Elaborado pelo autor.

A Figura 8 ilustra o carregamento do cenário *Wi-Fi Cracking* pelo modo Praticar. Nota-se a ausência da indicação do passo, do comando e da explicação, assim como seria o comportamento do Terminal real. Por se tratar de uma prática a nível intermediário, considera-se que o usuário ainda possa se confundir quanto aos comandos, por isso é possível digitar o comando `show next` para visualizar o comando esperado.

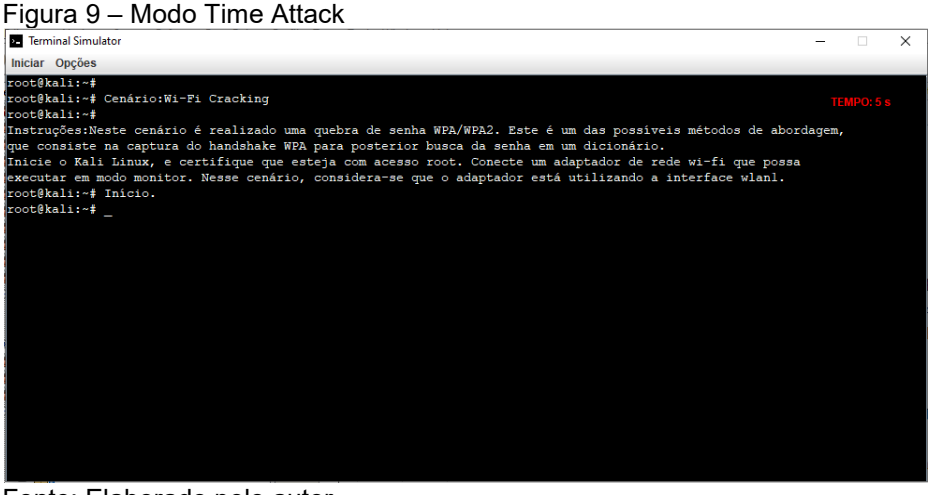
Figura 8 – Modo Praticar



Fonte: Elaborado pelo autor.

A Figura 9 ilustra o carregamento do cenário *Wi-Fi Cracking* pelo modo *Time Attack*. Nota-se ser semelhante ao modo Praticar, contudo, no canto superior direito, há a presença de um contador em contagem regressiva, representando o tempo restante que o usuário possui para digitar corretamente o comando esperado. Esse tempo é configurado durante a criação dos passos. Esse é um modo para usuários avançados que buscam desafiar seus conhecimentos e habilidades.

Figura 9 – Modo Time Attack



Fonte: Elaborado pelo autor.

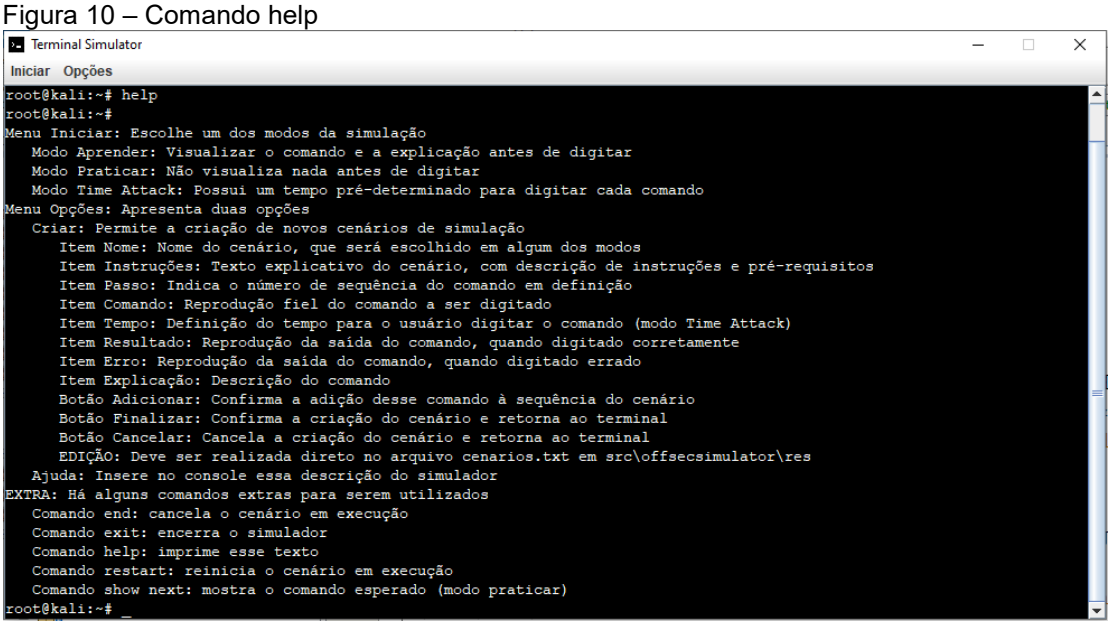
A condução dos cenários segue o fluxograma representado na Figura 5. Cada execução correta do comando esperado avança o cenário para o próximo passo, resultando na mensagem esperada. Um comando errado imprime uma mensagem de erro e mantêm o cenário no passo atual.

Há outros comandos pré-configurados que o usuário pode digitar para prover maior interação com o simulador:

- *end*: cancela o cenário em execução;
- *exit*: encerra o simulador;
- *help*: imprime um texto explicativo sobre o simulador;
- *restart*: reinicia o cenário em execução;
- *show next*: mostra o comando esperado (modo Praticar).

A execução do comando `help` pode ser realizada tanto diretamente na linha de comando, quanto acessando o menu “Opções” seguido por “Ajuda”, afinal, inicialmente o usuário não saberá a existência do comando `help`. O resultado da execução desse comando é demonstrado na Figura 10.

Figura 10 – Comando help

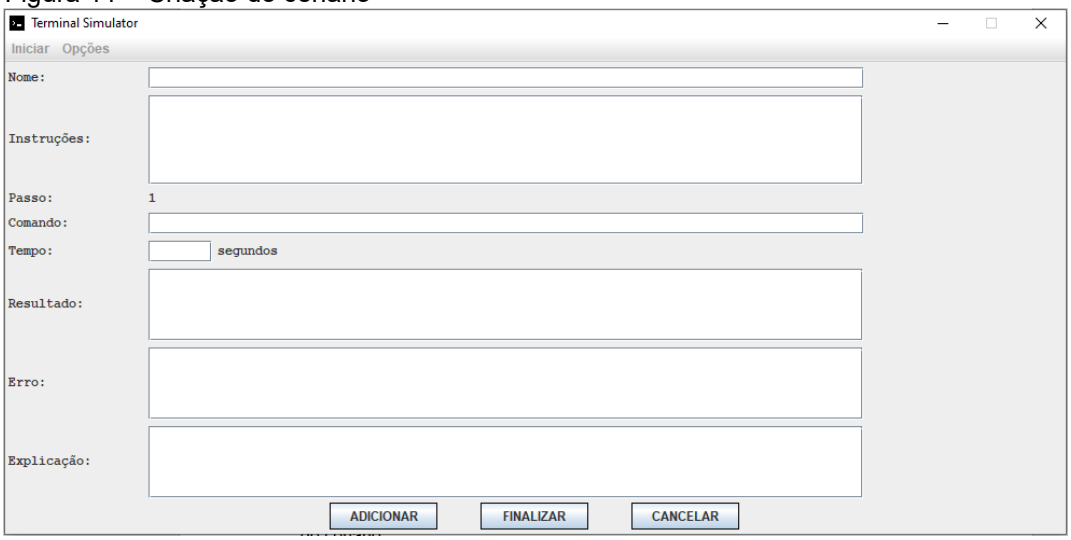


Fonte: Elaborado pelo autor.

Ao selecionar o menu “Opções” e em seguida “Criar”, abrirá a tela de criação de cenários de simulação, conforme Figura 11. Nessa tela é inserido o nome do cenário (que será incluído na listagem dos modos de simulação) e instruções gerais sobre o cenário. Em seguida são inseridos, para o primeiro passo, o comando, tempo de execução do mesmo em modo *Time Attack*, resultado a ser exibido pelo comando,

mensagem de erro quando digitado errado, e explicações desse passo. Deve-se pressionar o botão ADICIONAR para confirmar a criação do cenário e do primeiro passo. Em seguida, as opções de Nome e Instruções não ficarão mais editáveis, a contagem de passo incrementará automaticamente, e será possível inserir os dados do próximo passo. Essa iteração pode ser repetida quantas vezes for necessário, sempre confirmando com o botão ADICIONAR. Quando não houver mais passos a serem criados, o botão FINALIZAR deve ser pressionado. Será retornado ao terminal e o cenário será automaticamente adicionado à lista dos modos. O botão CANCELAR pode ser pressionado a qualquer momento, e a criação do cenário será cancelada, retornando ao terminal. Na Seção 4.2 é demonstrado passo-a-passo a criação de um cenário.

Figura 11 – Criação do cenário

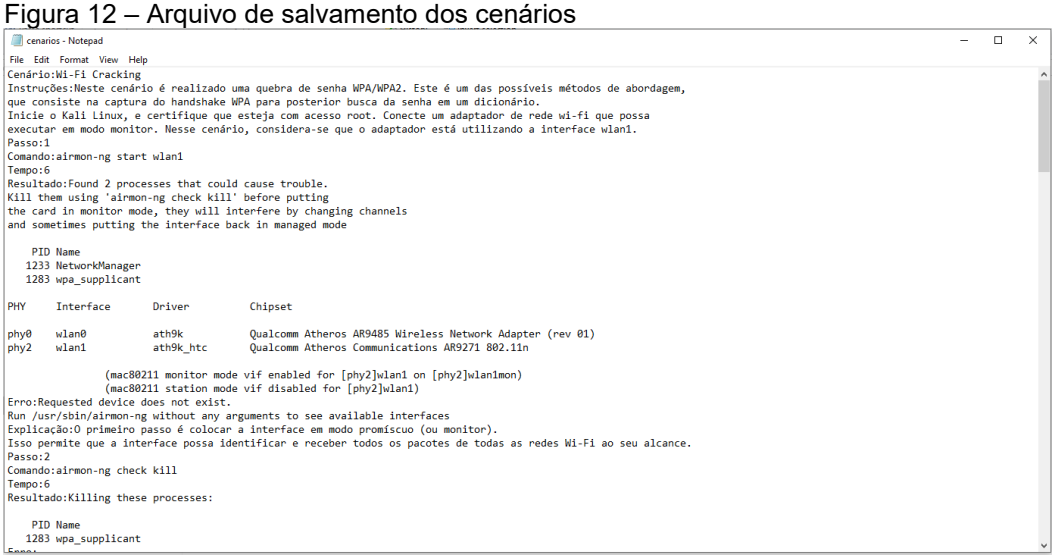


Fonte: Elaborado pelo autor.

Nesta tela de criação de cenário, os menus permanecem desabilitados. Somente após a conclusão ou cancelamento da criação dos cenários, que os menus são reativados.

Os cenários são salvos em um arquivo nomeado *cenarios.txt*, dentro do projeto do código-fonte. Quando o simulador é iniciado, automaticamente é carregado os cenários salvos nesse arquivo. A Figura 12 ilustra o início do cenário *Wi-Fi Cracking* salvo no arquivo.

Figura 12 – Arquivo de salvamento dos cenários



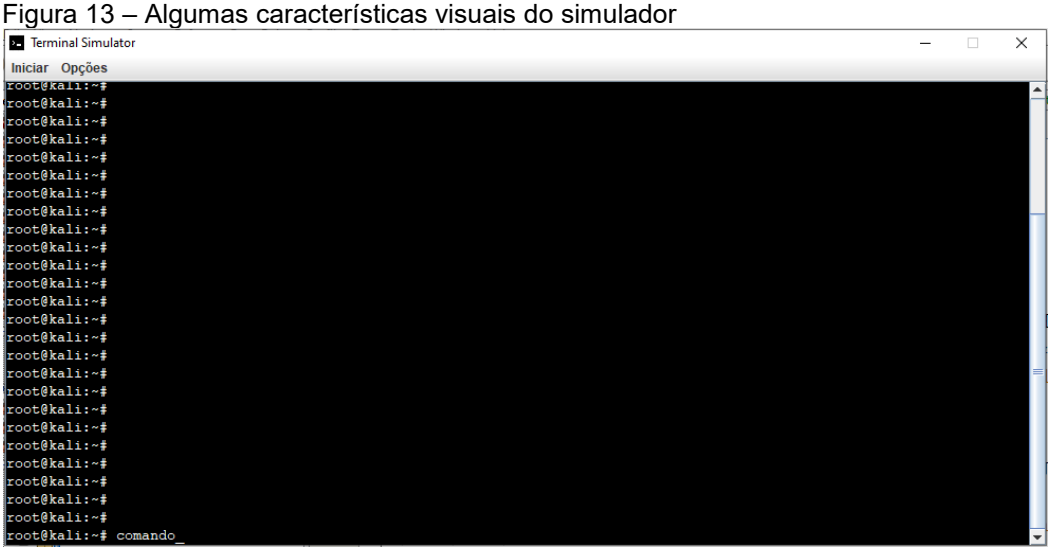
Fonte: Elaborado pelo autor.

Há uma ótima razão para se utilizar esse arquivo para guardar os cenários criados. Uma vez que esses cenários são reflexos da execução no terminal real, é conveniente que estes sejam copiados diretamente dessa execução. Isso pode ser feito com um procedimento de copiar o texto do terminal e colar em um bloco de notas, por exemplo. Dessa forma, pode ser mais prático colar esse texto diretamente no arquivo *cenarios.txt*, apenas assegurando que seja respeitado a estrutura a ser seguida. Para isso, basta observar a sequência, os comandos e separações entre os passos e cenários.

De forma similar, é mais prático a edição diretamente no arquivo texto, caso deseje alterar o tempo do contador, ou ajustar alguma mensagem. Por isso, após a familiarização com o uso desse arquivo, se torna muito mais prático sua manipulação direta do que o uso da interface de criação de cenários.

Outras *features* foram incluídas no comportamento do simulador. Se o número de linhas exceder o tamanho da janela, é criado uma barra de rolagem, semelhante ao Terminal do Linux. Ao pressionar ENTER, em uma linha vazia, o cursor prossegue para uma nova linha abaixo. Aplicou-se um efeito interessante no final de cada linha, apresentando o sinal de *underline* piscando. Esse sinal aparece por 500 milissegundos e desaparece por outros 500 milissegundos. A Figura 13 demonstra essas três características.

Figura 13 – Algumas características visuais do simulador



Fonte: Elaborado pelo autor.

O desenvolvimento dessa ferramenta resultou em mais de 1000 linhas de códigos digitados exclusivamente para esse trabalho, representando um projeto inteiramente original.

Dentre as diversas classes disponibilizadas pela linguagem Java, há o uso de: JFrame, JPanel, JMenu, JTextArea, JLayeredPane, GridBagConstrains, Document, Action, Timer, BufferedReader, ArrayList, Exception, entre outras. A criação do *layout* da aplicação é inteiramente desenvolvida em código, sem recorrer ao uso de *Graphical User Interface* (GUI) auxiliar para criação de interfaces, como o provido pela *Integrated Development Environment* (IDE) NetBeans. Diversos métodos dessas classes entre outros criados exclusivamente para a aplicação foram utilizados nesse projeto.

3.1 CONSIDERAÇÕES DO CAPÍTULO

O desenvolvimento da ferramenta foi conduzido de forma a assemelhá-la visualmente e funcionalmente ao Terminal do Linux, simulando o seu comportamento básico. O resultado obtido foi satisfatório e condiz com uma representação verossímil da ferramenta real, dentro do escopo adotado para esse trabalho.

Esse projeto continua em aperfeiçoamento, pois acredita-se no potencial de evolução da ferramenta. O resultado obtido até aqui, conforme imagens e discussões no capítulo, demonstra a facilidade e praticidade no uso do simulador, atingindo os objetivos desejados.

4. ESTUDO DE CASO – ATAQUE À REDE SEM FIO

Ataques às redes sem fio visam explorar fragilidades dos protocolos de segurança, das senhas ou dos roteadores Wi-Fi e *Access Points* (APs), a fim de obter acesso não autorizado à rede. A ampla adoção de redes sem fio em ambientes empresariais e residenciais despertou a atenção de interessados em investigá-las, fomentando o surgimento de ferramentas e meios para explorar suas vulnerabilidades. Nesse contexto, auditar redes sem fio é uma tarefa fundamental para a segurança da infraestrutura de comunicação. Para isso, as mesmas ferramentas utilizadas para a invasão maliciosa podem ser usadas para detectar fragilidades e auxiliar nas correções de segurança.

O Kali Linux apresenta nativamente a inclusão do Aircrack-ng (Aircrack-ng, 2020), que não é uma única ferramenta, mas uma suíte de ferramentas e *scripts* direcionados a atividades específicas, utilizados para, em conjunto, explorar fragilidades (“crackear”) de redes sem fio. A ferramenta original, Aircrack, foi criada em 2010, sendo uma das mais antigas e que continuou evoluindo até os dias atuais, se tornando uma ferramenta de referência para essa finalidade.

O termo “ng” corresponde a “*new generation*” (“nova geração”, em tradução livre), sendo um *fork* do projeto inicial, Aircrack, que não é mais suportado pela comunidade. A versão atual é 1.6, datada de janeiro de 2020. A Figura 14 ilustra o logotipo da suíte Aircrack-ng.

Figura 14 – Aircrack-ng: logotipo

Fonte: Aircrack-ng, 2020.

Dentre as ferramentas da suíte Aircrack-ng há a ferramenta de mesmo nome, aircrack-ng, usada para “crackear” as senhas das redes Wi-Fi. Muitas outras ferramentas auxiliam em outras atividades e ataques específicos, mas, no geral, são comumente utilizadas as seguintes ferramentas: airmon-ng para deixar a placa de rede em modo promíscuo e capturar pacotes de todas as redes Wi-Fi ao alcance;

airodump-ng para capturar pacotes conforme especificado (focando em uma rede específica); e aireplay-ng para gerar um tráfego injetando pacotes na comunicação. Na Seção 4.1 é demonstrado o uso dessas ferramentas.

Para a condução de uma tarefa de quebra de senhas Wi-Fi são necessários alguns equipamentos. Além de um roteador alvo pré-configurado para o teste de invasão, é necessário um adaptador de redes *wireless* compatível (NULL BYTE, 2019). O adaptador usado neste trabalho foi o TP-LINK TL-WN722N. O roteador utilizado para ter sua rede Wi-Fi “crackeada” foi o D-LINK DI-524.

Em função da necessidade de equipamentos extras, esta pode ser uma atividade mais difícil de ser praticada e demonstrada em um ambiente de ensino. Dessa forma, se torna um cenário ideal para ser simulado.

4.1 CENÁRIO REAL

A seguir são demonstrados os passos de condução de um cenário real para ser realizado uma quebra de senha WPA/WPA2. WPA é a sigla para *Wi-Fi Protected Access*, um padrão de segurança certificado pela Wi-Fi Alliance (WI-FI ALLIANCE, 2020). O WPA2 foi introduzido em 2004 e tem sido um padrão dominante desde então. Apenas em 2018 foi anunciada a evolução desse padrão, o WPA3, que ainda está em início de adoção.

Destaca-se que esse cenário apresenta um dos possíveis métodos de abordagem, que consiste na captura do *handshake WPA* para posterior busca da senha em um dicionário. A condução desse cenário é importante para se obter as mensagens de tela a serem inseridas na simulação.

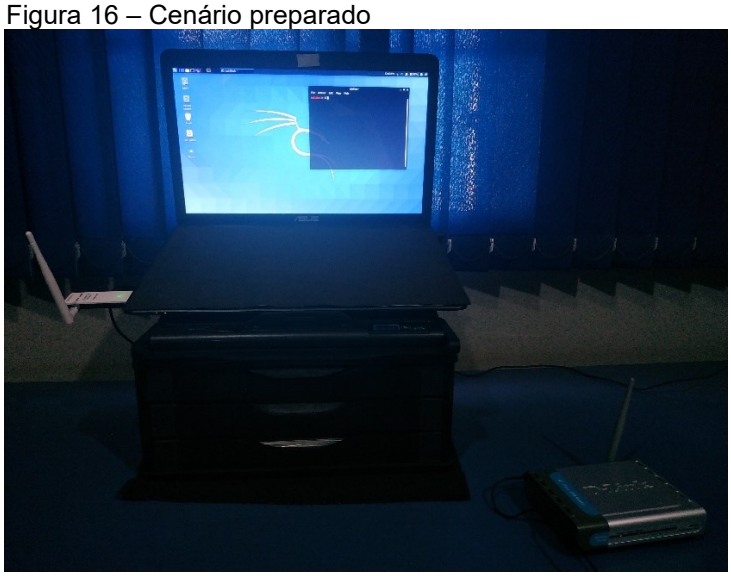
A Figura 15 demonstra a configuração do roteador utilizado nesse cenário, com a tela de configuração (endereço 192.168.0.1) aberta no navegador, evidenciando que se trata de uma rede sob responsabilidade do autor. O nome da rede é “rede_alvo” e sua senha, a ser obtida, é “umasenhacomum”.

Figura 15 – Configuração da rede Wi-Fi

Fonte: Elaborado pelo autor.

A Figura 16 apresenta o cenário pronto para início da execução. O roteador está operando a rede Wi-Fi configurada, o notebook está executando o Kali Linux versão 2020.2, com um terminal aberto. E o adaptador USB de rede Wi-Fi está conectado ao notebook.

Figura 16 – Cenário preparado



Fonte: Elaborado pelo autor.

Desde a versão 2020.1, o Kali Linux não apresenta mais usuário *root* padrão. Isso significa que alguns comandos e ferramentas não funcionam por padrão, sendo necessário escalar o privilégio de acesso. Nesse trabalho, pelo fato do Kali Linux ser executado a partir de um Live DVD, foi utilizado o comando a seguir para garantir acesso à todas as ferramentas da distribuição:

```
sudo su
```

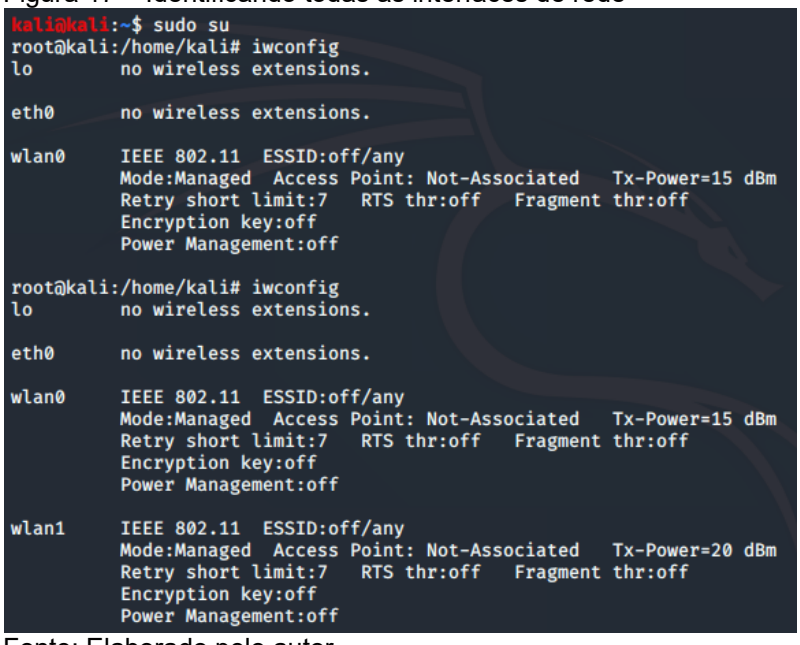
A seguir serão apresentados e numerados seis passos essenciais que compõem as etapas para se “crackear” uma senha WPA/WPA2 de uma rede Wi-Fi. Esses passos são registrados para posteriormente serem representados na ferramenta de simulação. Alguns passos adicionais são apontados para melhor ilustração da condução desse cenário.

Inicialmente, o usuário pode conferir os adaptadores de rede reconhecidos pelo sistema operacional. Isso é feito com o comando:

```
iwconfig
```

A Figura 17 ilustra que na primeira execução desse comando, o adaptador não estava conectado na porta USB, mas na segunda execução, com o adaptador conectado, ele foi reconhecido e associado à interface wlan1.

Figura 17 – Identificando todas as interfaces de rede



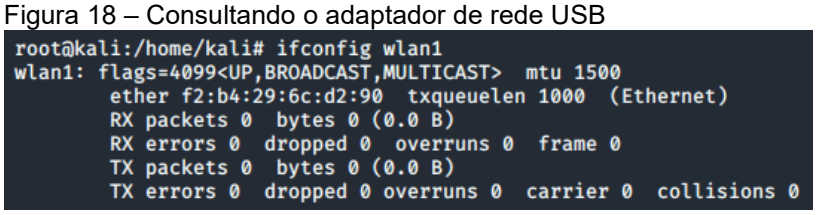
Fonte: Elaborado pelo autor.

Ainda é possível consultar diretamente a interface para mais detalhes. Para isso deve ser digitado o comando:

```
ifconfig wlan1
```

A execução desse comando gerará um resultado semelhante ao da Figura 18.

Figura 18 – Consultando o adaptador de rede USB



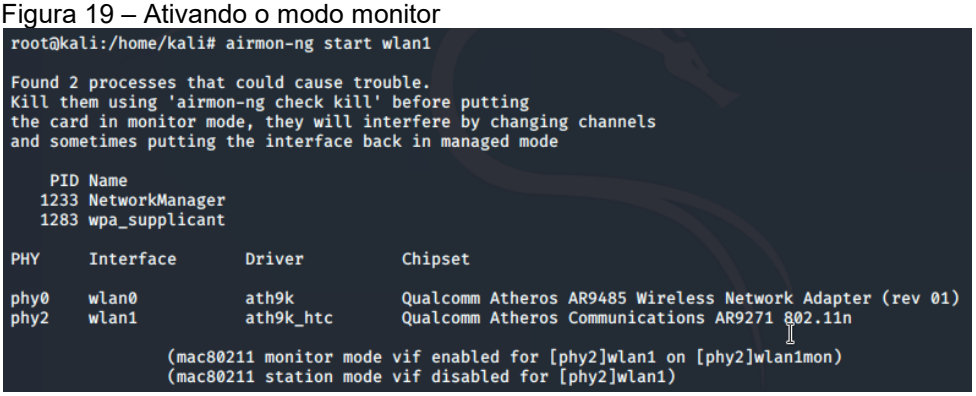
Fonte: Elaborado pelo autor.

O primeiro passo (PASSO 1) é colocar a interface em modo promíscuo (ou monitor). Isso permite que a interface possa identificar e receber todos os pacotes de todas as redes Wi-Fi ao seu alcance. Inicia-se esse modo com o comando:

```
airmon-ng start wlan1
```

A Figura 19 demonstra o possível resultado dessa execução.

Figura 19 – Ativando o modo monitor



Fonte: Elaborado pelo autor.

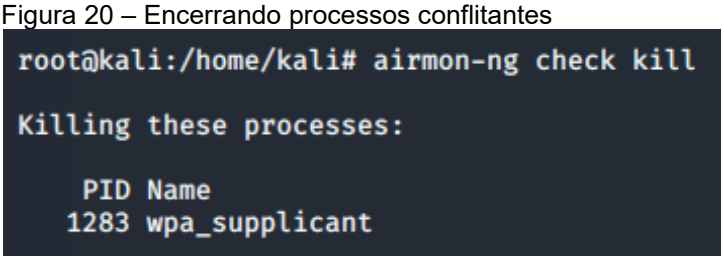
Observa-se a mensagem ao final indicando que a interface está *enabled* para modo *monitor*, indicando que este modo responde na interface *wlan1mon*.

Contudo, uma mensagem de aviso foi gerada, indicando possível interferência de outros processos, que precisam ser encerrados, representando o segundo passo (PASSO 2) a ser realizado. Isso é feito com o comando:

```
airmon-ng check kill
```

A execução desse comando poderá gerar alguma atividade no adaptador, com sua luz indicativa começando a piscar. O resultado dessa execução é visualizado na Figura 20.

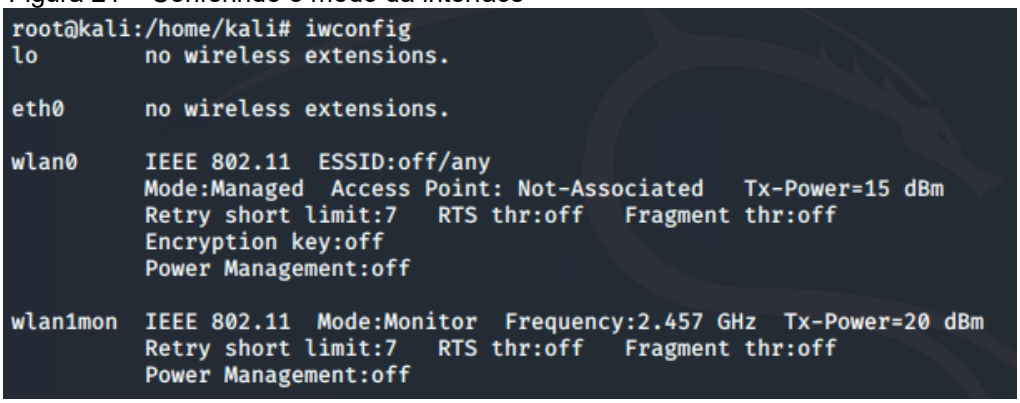
Figura 20 – Encerrando processos conflitantes



Fonte: Elaborado pelo autor.

Para confirmar a mudança de modo, pode-se consultar novamente as interfaces. Nesse momento, a interface em modo monitor está ativa, conforme indicado na Figura 21.

Figura 21 – Conferindo o modo da interface



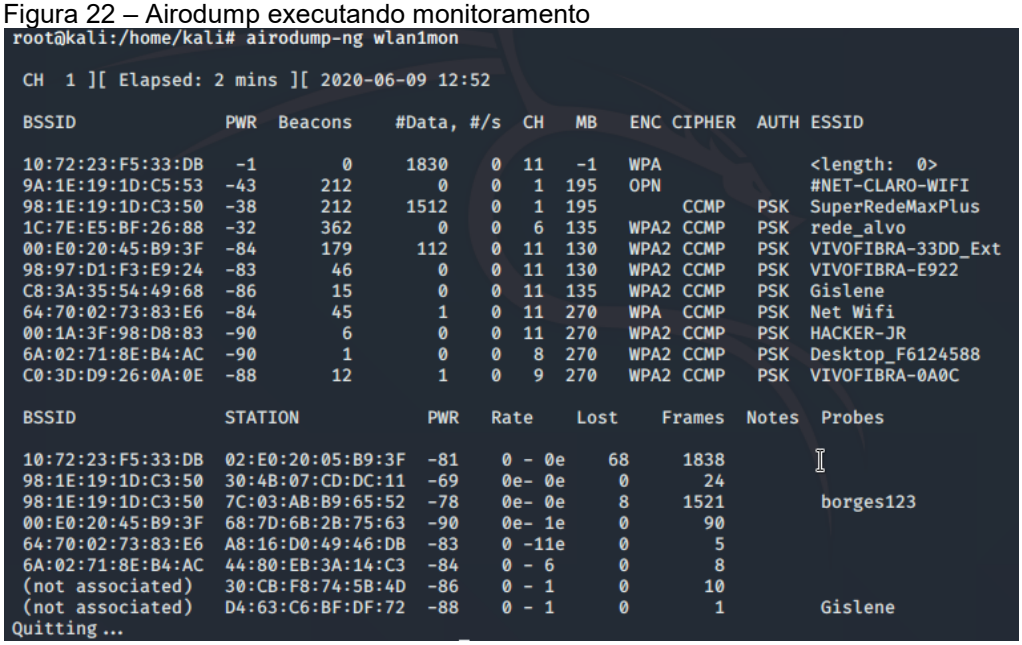
Fonte: Elaborado pelo autor.

O terceiro passo (PASSO 3) é iniciar um monitoramento do ambiente para identificar todas as redes Wi-Fi ao alcance. Com isso serão identificados os APs e as respectivas informações associadas (BSSID, endereço MAC, *hosts* conectados, canal em uso, etc.). Para isso, deve ser executado:

```
airodump-ng wlan1mon
```

O resultado é dinâmico, uma vez que varia conforme a mudança da potência dos sinais, associação e desassociação de *hosts* das redes, entre outras atividades. A Figura 22 ilustra um instante desse monitoramento.

Figura 22 – Airodump executando monitoramento



Fonte: Elaborado pelo autor.

A Figura 22 é importante pois nessa tela são consultadas algumas informações da rede a ser alvo do ataque. Nesse caso, será atacada a rede “rede_alvo”. Para isso é necessário anotar o canal (CH) e o BSSID. Esse monitoramento é encerrado com as teclas CTRL + C pressionadas.

Em seguida, o quarto passo (PASSO 4) é iniciar um monitoramento na rede que foi identificada para ser atacada. Para isso é utilizado o comando:

```
airodump-ng -c 6 --bssid 1C:7E:E5:BF:26:88 -w
/home/kali/Documents/rede_alvo wlan1mon
```

No comando anterior, em -c <channel>, channel é o número do canal em que está operando a rede Wi-Fi. Em --bssid <bssid>, bssid é o endereço MAC (BSSID) do AP. Em -w <caminho>, caminho indica onde os arquivos desse monitoramento serão salvos. É recomendado utilizar o nome da rede (nesse caso, “rede_alvo”). Serão gerados cinco arquivos tendo como prefixo este nome. Por fim, é indicado a interface em modo monitor.

A execução apresenta uma tela dinâmica, indicando os dispositivos conectados na rede e a quantidade de pacotes capturados, conforme ilustrado pela Figura 23. Não é de grande utilidade monitorar uma rede sem *hosts*, pois será necessário a presença de ao menos um *host* para se capturar o tráfego da rede e um *handshake*.

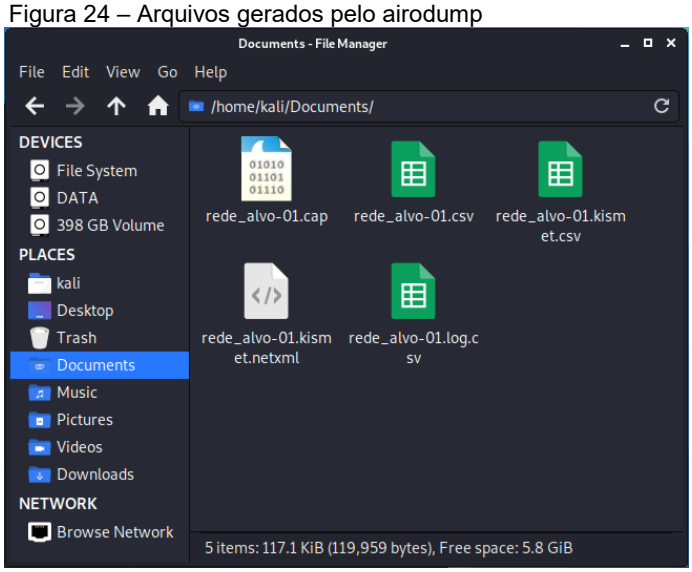
Figura 23 – Airodump executando em uma rede específica

```
CH  6 ][ Elapsed: 12 s ][ 2020-06-09 13:00

BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

1C:7E:E5:BF:26:88  -22 100      127       11    0   6  135   WPA2 CCMP   PSK  rede_alvo

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

1C:7E:E5:BF:26:88  94:87:E0:53:18:5F  -30    0 - 1e     0       24
```

Fonte: Elaborado pelo autor.

Enquanto esse monitoramento ocorre, são gerados arquivos no diretório indicado, como representado na Figura 24.

Figura 24 – Arquivos gerados pelo airodump



Fonte: Elaborado pelo autor.

Como mencionado anteriormente, o objetivo desse monitoramento é obter o *handshake* de conexão, que juntamente com os dados da rede e tráfego capturado, poderá ser confrontado com um dicionário (*wordlist*) para identificar qual seja a senha da rede.

O *handshake* pode ser obtido durante a autenticação de um usuário na rede, porém, isso pode ser demorado. Para acelerar o processo é possível “derrubar” um usuário da rede (desautenticá-lo) enviando uma mensagem falsa, para que em seguida ele se reconecte e seja possível capturar o *handshake*.

O quinto passo (PASSO 5) consiste em abrir um novo terminal (é importante que continue monitorando a rede pelo terminal anterior) e digite:

```
aireplay-ng -0 1 -a 1C:7E:E5:BF:26:88 -c 94:87:E0:53:18:5F wlan1mon
```

A Figura 25 ilustra o resultado da execução desse commando.

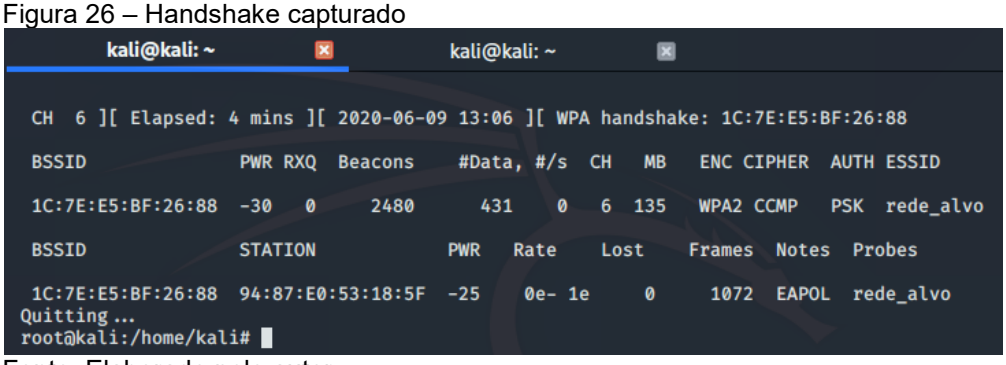
Figura 25 – Aireplay executando deauth

Fonte: Elaborado pelo autor.

No comando acima, em “-0” é indicado que será feito um ataque de desassociação de cliente (deauth). “1” indica que será enviado apenas um pacote (nesse caso só há um cliente na rede). Em -a <mac_ap>, mac_ap indica o endereço MAC do AP, e em -c <mac_cli>, mac_cli indica o endereço MAC do cliente. Por fim, é indicado a interface em modo monitor.

Logo após a execução do aireplay, deve-se retornar ao terminal anterior que estava em monitoramento da rede alvo, e aguardar alguns instantes. O usuário da rede foi desconectado dela, e é provável que se reconecte à rede, nesse caso, permitindo que capture o *handshake*. Isso será confirmado com a mensagem no canto superior direito da tela: “WPA handshake”. A Figura 26 demonstra a captura do *handshake*. Após essa captura, pode-se encerrar o monitoramento (CTRL + C).

Figura 26 – Handshake capturado



Fonte: Elaborado pelo autor.

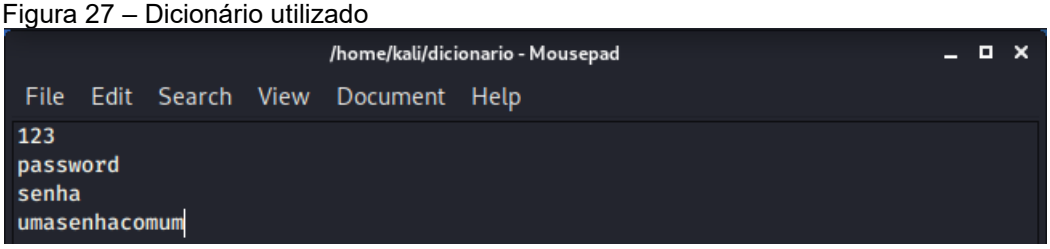
Com os dados necessários capturados, pode-se avançar para o sexto e último passo (PASSO 6). Nessa etapa inicia-se a tentativa de quebra da senha. Para isso é utilizado um dicionário (*wordlist*) que deve conter uma série de possíveis senhas.

Pode-se utilizar tanto um dicionário existente, genérico, quanto pode ser criado um próprio dicionário. Em ataques mais organizados e específicos, monta-se e utiliza-

se um dicionário direcionado para a rede/usuário alvo. Programas como Crunch, Cewl e Cupp auxiliam nessa tarefa.

A Figura 27 apresenta o dicionário utilizado nesse cenário. É um exemplo didático que contém apenas quatro palavras de possíveis senhas, sendo que a última corresponde à senha da rede.

Figura 27 – Dicionário utilizado



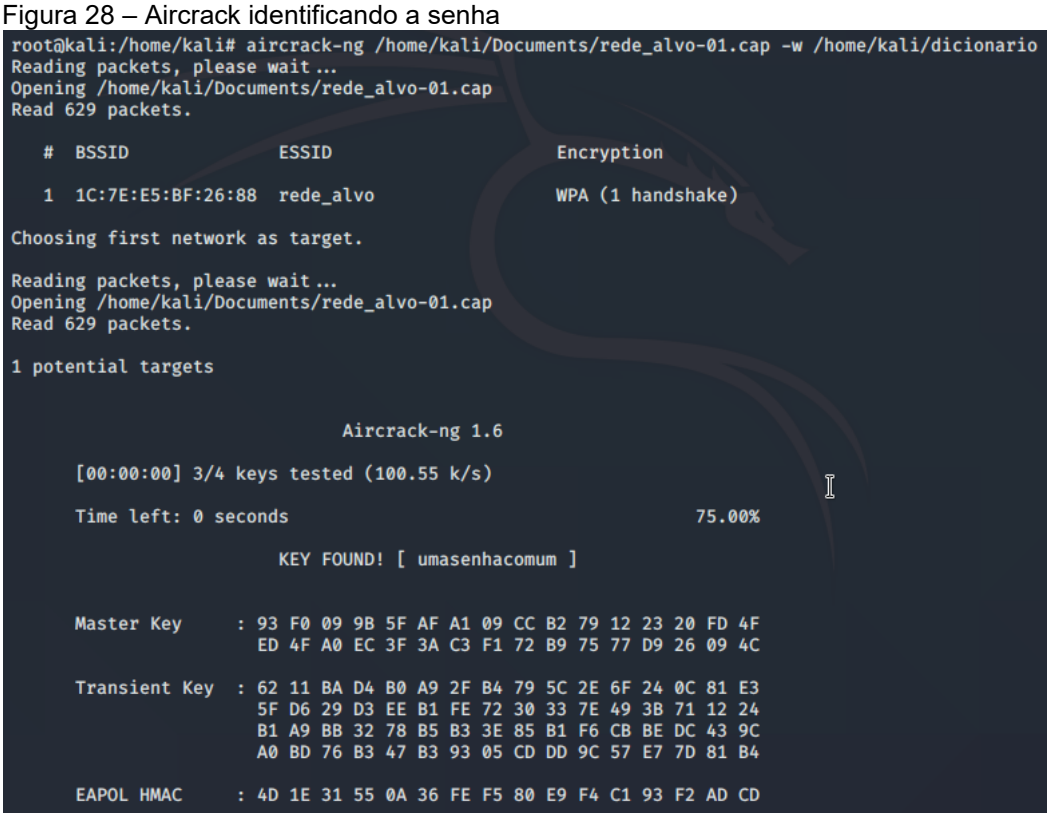
Fonte: Elaborado pelo autor.

Com o dicionário criado, executa-se o comando:

```
aircrack-ng /home/kali/Documents/rede_alvo-01.cap -w /home/kali/dicionario
```

A Figura 28 representa o sucesso da execução, quando a senha da rede foi encontrada no dicionário.

Figura 28 – Aircrack identificando a senha



Fonte: Elaborado pelo autor.

No comando acima, inicialmente é indicado o caminho para o arquivo .cap gerado pelo monitoramento da rede. E depois é indicado o dicionário que será utilizado.

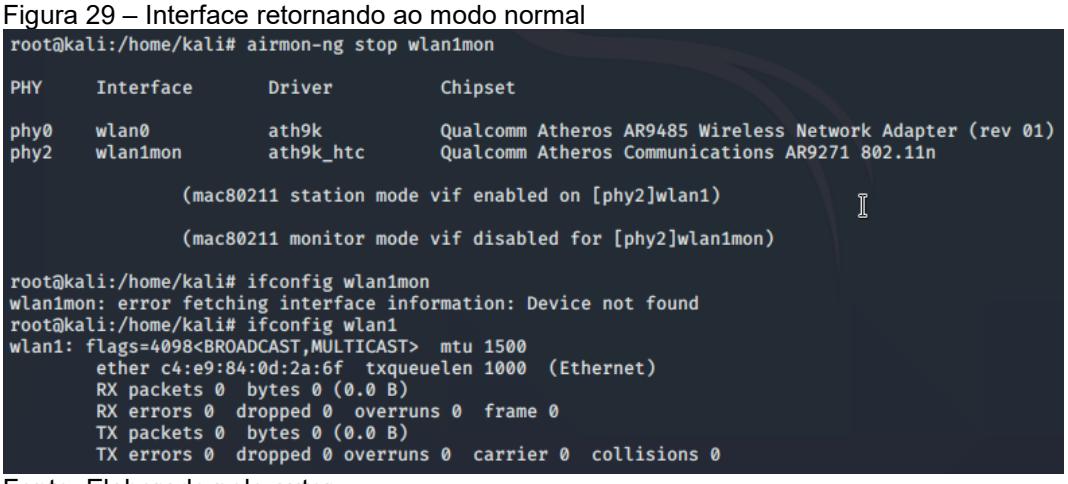
Percebe-se que na Figura 28 foi encontrado a palavra (*key*) no dicionário, que, conforme indicado, é “umasenhacomum”, sendo essa portanto a senha da rede. Por esse exemplo, esse processo foi executado rapidamente (0 segundos) porque por fins didáticos esse dicionário possui poucas palavras. Numa situação real, pode-se demorar muito mais, e possivelmente nem identificar a senha, levando à necessidade de recriar o dicionário, ou buscar outros métodos de ataque.

Por fim, deve-se retornar a interface ao modo normal, desativando o modo monitor:

```
airmon-ng stop wlan1mon
```

A Figura 29 demonstra a posterior verificação da interface wlan1.

Figura 29 – Interface retornando ao modo normal



Fonte: Elaborado pelo autor.

4.2 CENÁRIO SIMULADO

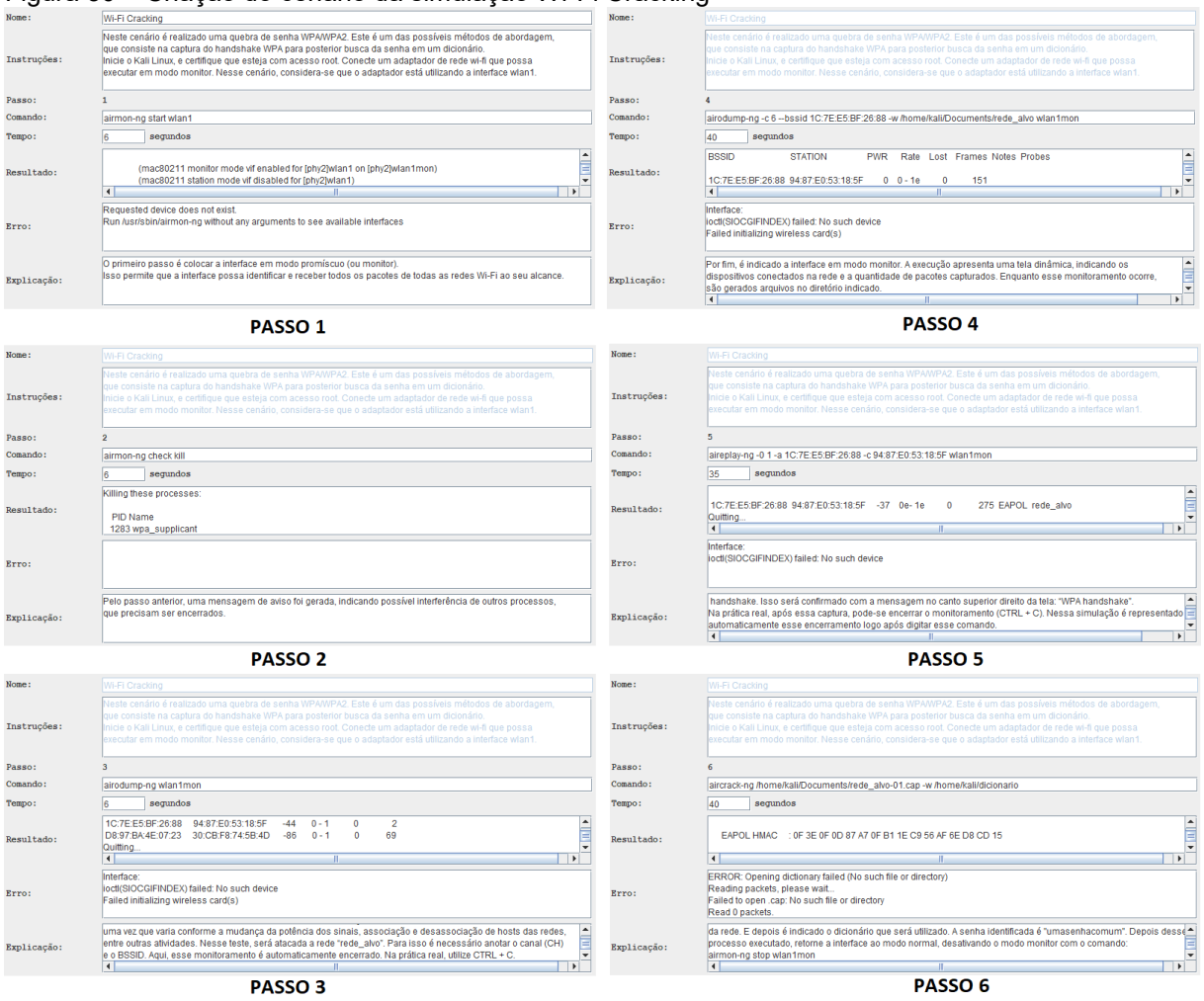
Após a condução do cenário real, foram coletados os resultados das execuções dos comandos, tanto corretamente executados quanto possíveis erros enfrentados. Os comandos, resultados corretos e erros, e mensagens explicativas conforme apresentadas na Seção 4.1, foram inseridos no simulador para a criação do cenário denominado *Wi-Fi Cracking*.

Observa-se que alguns dos passos executados não são essenciais para a execução do processo. Sendo assim, na simulação foram representados os seis passos essenciais, destacados na Seção 4.1, até a quebra da senha. Destaca-se também que diferentes erros podem surgir dependendo do equívoco na sintaxe do comando. Contudo, foi adotado apenas uma mensagem de erro mais comum que pode ocorrer ao digitar erroneamente o comando do passo simulado.

Outro ponto de abstração é o comando CTRL + C, cuja execução é representada automaticamente para o encerramento de alguns passos. Por fim, o simulador representa apenas um terminal, dessa forma, o quinto passo é representado na mesma tela. Todas essas observações são inseridas no item “Explicação” de cada passo.

A Figura 30 ilustra as 6 telas preenchidas para a criação desses seis passos.

Figura 30 – Criação do cenário da simulação Wi-Fi Cracking

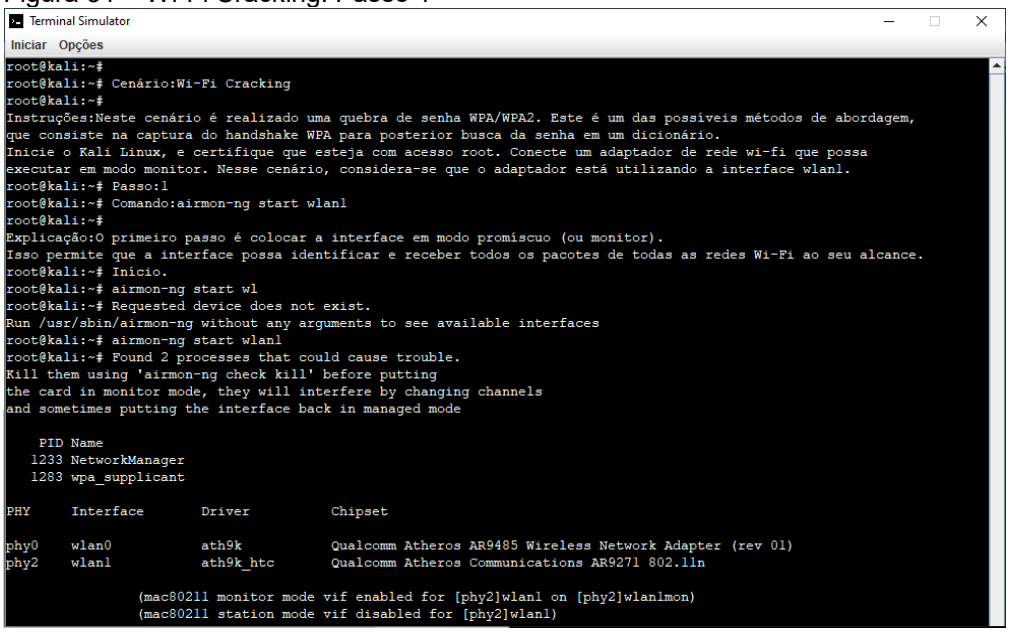


Fonte: Elaborado pelo autor.

Esse é um cenário base já acrescentado por padrão na ferramenta de simulação. A seguir serão apresentadas as telas de cada um desses passos, seguindo o fluxo do modo “Aprender”. Cada passo corresponde a execução de um comando errado, para visualizar a mensagem de erro, seguido pela execução do comando correto, para verificação do resultado esperado.

A Figura 31 demonstra a execução do primeiro passo do cenário *Wi-Fi Cracking*.

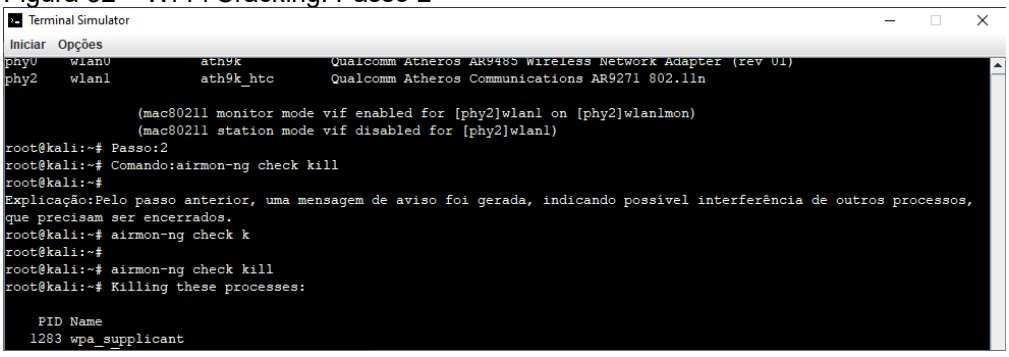
Figura 31 – Wi-Fi Cracking: Passo 1



Fonte: Elaborado pelo autor.

A Figura 32 demonstra a execução do segundo passo. Observa-se que a execução errada do comando não resulta em uma mensagem de erro. Isso foi abstraído da observação do caso real.

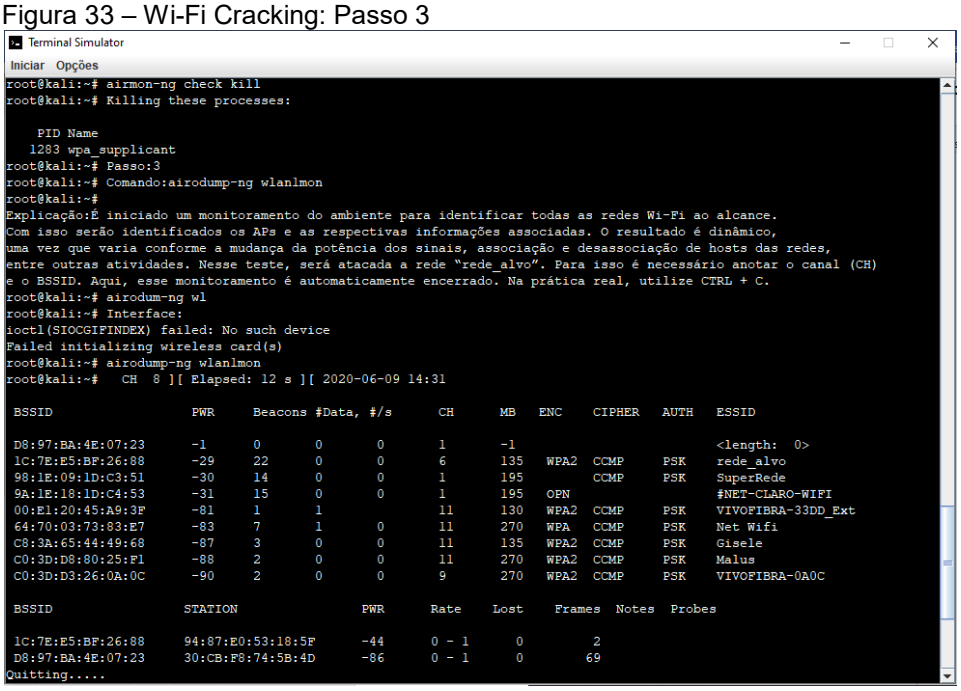
Figura 32 – Wi-Fi Cracking: Passo 2



Fonte: Elaborado pelo autor.

A Figura 33 demonstra a execução do terceiro passo.

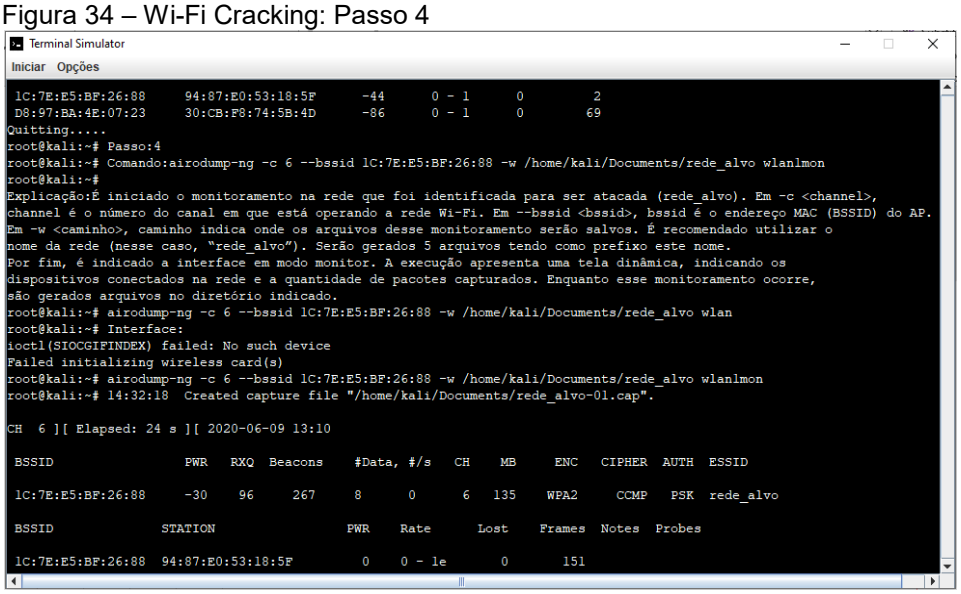
Figura 33 – Wi-Fi Cracking: Passo 3



Fonte: Elaborado pelo autor.

A Figura 34 demonstra a execução do quarto passo.

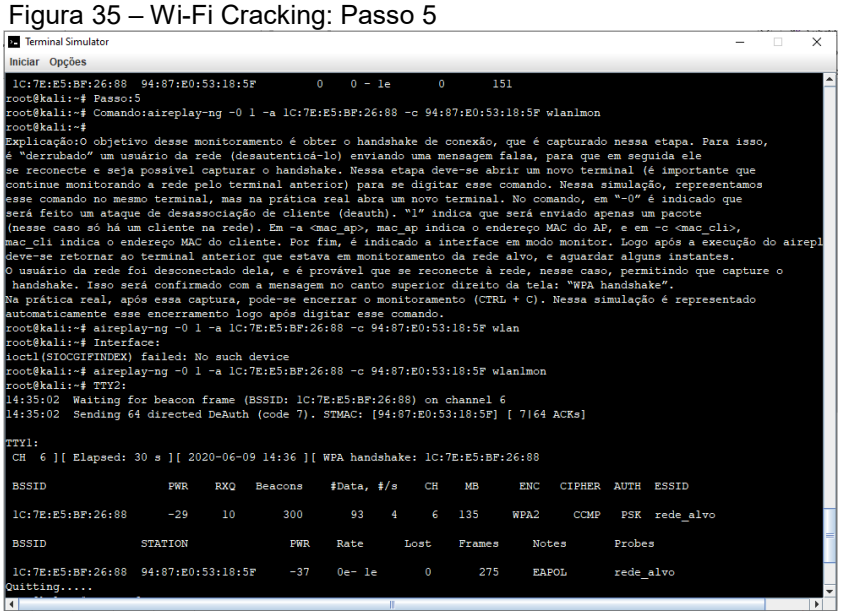
Figura 34 – Wi-Fi Cracking: Passo 4



Fonte: Elaborado pelo autor.

A Figura 35 demonstra a execução do quinto passo.

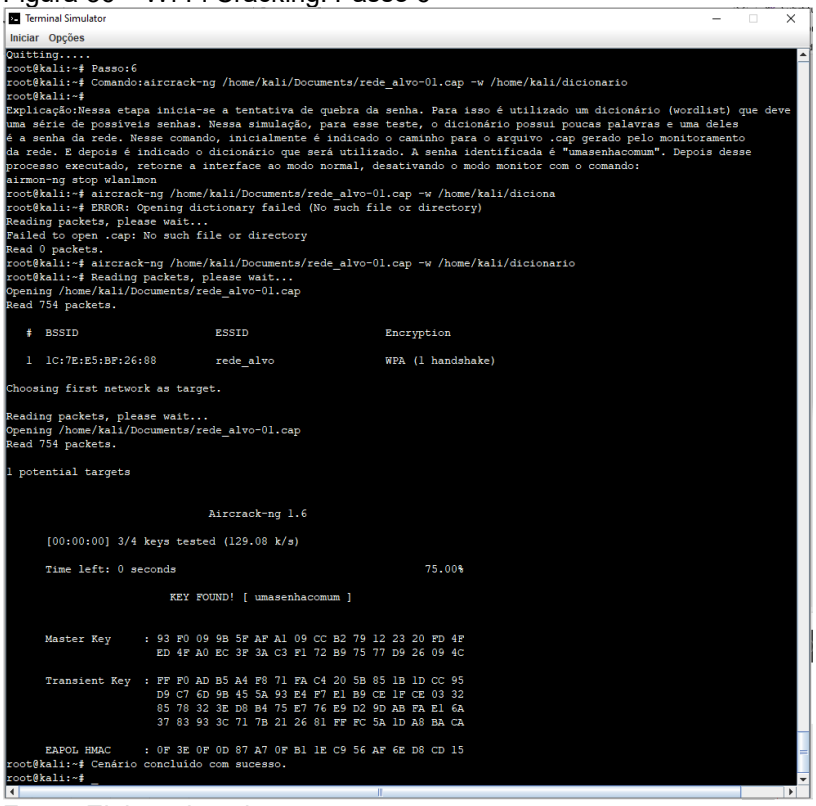
Figura 35 – Wi-Fi Cracking: Passo 5



Fonte: Elaborado pelo autor.

A Figura 36 demonstra a execução do sexto e último passo.

Figura 36 – Wi-Fi Cracking: Passo 6



Fonte: Elaborado pelo autor.

4.3 CONSIDERAÇÕES DO CAPÍTULO

A ferramenta desenvolvida demonstrou a capacidade de simular de forma verossímil o Terminal do Kali Linux. Todos os detalhes da ferramenta foram apresentados no Capítulo 3.

Evidencia-se como o fluxo de sequência de comandos na ferramenta se comporta da mesma forma que no Terminal. As diferenças perceptíveis são quanto ao acréscimo intencional de mensagens instrutivas e ao texto referente ao usuário e diretório mantido de forma mais “limpa” (root@kali:~#). Nesse cenário em específico, observa-se a representação automática da execução do comando CTRL + C e da representação em um único Terminal de uma operação realizada em dois Terminais.

Ressalta-se que, como uma versão inicial, a ferramenta naturalmente é uma prova de conceito e passível de evolução para representações de interações mais complexas. No entanto, em sua versão atual, como evidenciado nesse capítulo, a ferramenta desenvolvida apresenta-se como um recurso factível para representação de execução de comandos em um Terminal.

Esse estudo de caso comprovou a vantagem do uso da ferramenta para fins didáticos. Uma vez com o cenário configurado e inserido na ferramenta, sua reprodução pode ser realizada a qualquer instante, de forma imediata e prática, sem a necessidade de novas configurações e preparações para cada nova repetição, em contraste a um cenário real.

5. ANÁLISE DO QUESTIONÁRIO

Um questionário foi aplicado para os alunos do curso de Segurança da Informação da FATEC Americana a fim de identificar os conhecimentos e interesse dos alunos em segurança ofensiva. O questionário encontra-se no Apêndice A e as 45 respostas obtidas encontram-se no Apêndice B.

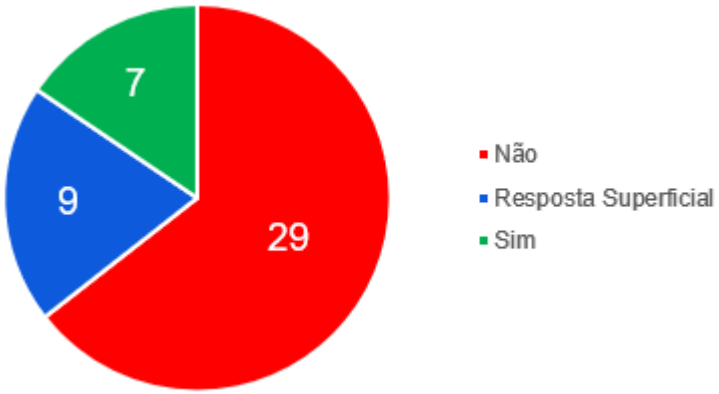
Observando as respostas da primeira questão, referente ao momento do aluno no curso, foi observado que a maioria dos respondentes se encontra no último ano do curso, conforme indicado no Gráfico 1. Isso representa um perfil de aluno maduro dentro da evolução do curso, tendo cumprido diversas disciplinas da grade curricular.

Gráfico 1 - Situação dos participantes perante o curso de SI

Fonte: Elaborado pelo autor.

A terceira questão busca identificar o nível de conhecimento do respondente quanto à segurança ofensiva em prática. As respostas foram compiladas em três categorias: Sim, para os que demonstram algum conhecimento à respeito de uso de ferramentas de segurança ofensiva; Não, para os que reconheceram não saber ou não se lembrar; Resposta Superficial, para os que disseram saber, mas não demonstram, conforme pedido no enunciado na questão. O Gráfico 2 apresenta essa compilação.

Gráfico 2 - Conhecimento dos participantes sobre segurança ofensiva



Fonte: Elaborado pelo autor.

O primeiro ponto de destaque é quanto à falta de familiaridade dos respondentes quanto à prática de segurança ofensiva, o que fortalece a proposta de uma ferramenta lúdica para essa finalidade de ensino, como a apresentada nesse trabalho. Outro ponto é quanto à confusão observada em relação ao termo “segurança ofensiva”. Apesar de uma breve explicação na introdução do questionário, alguns participantes responderam positivamente, mas se referindo a identificar sites ou e-mails maliciosos em nível de usuário, enquanto outros mencionaram atividades de monitoramento de tráfego de rede, que não caracterizam ações de segurança ofensiva. Muitos também mencionaram algumas ferramentas e conceitos, mas sem demonstrar familiaridade em seu uso. Por isso, todas essas respostas foram categorizadas como “Resposta Superficial”, uma vez que a questão buscava identificar real conhecimento prático do respondente, conforme explícito em seu enunciado.

Por fim, 7 respostas (15% dos participantes) foram identificadas com traços de real conhecimento do respondente. Entre essas respostas, encontra-se menções à *brute force* para quebra de senha, injeção de SQL e captura de tráfego de rede. Dentre essas respostas, 4 demonstram mais conhecimentos variados, sendo que se percebe que um desses participantes já trabalha na área de segurança da informação. Mesmo entre esses respondentes, percebe-se menções em relação à identificação de ataques e proteção da rede, o que não é uma ação de segurança ofensiva, que se trata de realizar o ataque em si.

A quarta questão visa instigar o aluno quanto à ferramenta proposta nesse trabalho, a fim de identificar o interesse pelo público-alvo ao qual ela se destina. A receptividade foi grande, com 44 respostas positivas entre os 45 participantes. O único aluno que respondeu não se interessar pela ferramenta também demonstrou não possuir nenhum conhecimento sobre segurança ofensiva, portanto é um aluno que, apesar de estar no curso de Segurança da Informação, não conhece e nem deseja saber mais sobre segurança ofensiva.

A quinta pergunta do questionário visa colher sugestões dos participantes para a ferramenta de simulação. As sugestões foram compiladas nos seguintes pontos:

- Atualizações frequentes;
- Ferramenta simples, didática e acessível aos alunos;
- Não Linear (mais de uma resposta);
- Documentação, explicações, tutorial, manual, biblioteca de comandos, passo-a-passo;
- Multiplataforma;
- Proximidade com a realidade;
- Código-aberto;
- Inclusão da ferramenta em sala de aula;
- Gamificação.

Ressalta-se que os pontos acima podem ter sido mencionados por mais de um respondente. Destaca-se a menção frequente ao que se refere a alguma forma de orientar a execução dos comandos, o que foi sugerido por vários alunos de diversas maneiras. A ferramenta apresentada nesse trabalho possui esse intuito: fornecer um aprendizado ao aluno, destacando a sequência de passos a serem executados e descrevendo cada um deles.

Outro ponto também de destaque, foi a recomendação de cinco alunos sobre o uso de uma ferramenta dessa natureza em sala de aula, para aprendizado e prática inicial dessas atividades de segurança. Essa é uma das propostas desse trabalho, em moldar uma ferramenta que possa ser empregada em sala de aula para futuros alunos do curso de Segurança da Informação.

Outro ponto atendido pela ferramenta, que foi um fator norteador do trabalho, é a proximidade com a realidade. Sua interface gráfica foi desenvolvida para se assemelhar à do Terminal do Kali Linux, representando as mensagens da forma mais fidedigna possível.

Em relação à gamificação, um primeiro ponto já implantado foi o modo “Time Attack”, que permite ao aluno competir contra o tempo para execução dos comandos o mais rapidamente possível. Futuramente, a criação de um *ranking* e mecanismos de recompensa poderão ser implantados. Outros pontos como tornar a ferramenta multiplataforma e código-aberto estão sendo considerados para atualizações futuras.

Outros comentários identificados foram sobre a vantagem de ter uma ferramenta para “praticar sem uma zona de real risco”. De fato, a natureza dessa ferramenta provê um ambiente de testes sem riscos, sem expor o usuário ao desconhecido e ainda não dominado, e nem deixar seu próprio ambiente vulnerável. Outro aluno mencionou que “um dos maiores problemas era manter uma máquina virtual para testar essas ferramentas”, e realmente essa ferramenta não possui a necessidade de configurações extras de máquinas virtuais e demais recursos do ambiente, sendo prático o início do cenário de interesse, o que se alinha com outro comentário, sobre uma ferramenta para “gerar agilidade” nesse processo de prática.

Outro comentário citou a “dificuldade em Linux”. Uma ferramenta como essa cria uma abertura amigável à inserção de alunos ainda não familiarizados com o ambiente. Sem exigir instalações, configurações, entre outros procedimentos, e colocando diretamente o aluno no cenário de prática, pode ser um fator motivador para encorajar mais alunos ao uso do Linux e ao desenvolvimento da prática de segurança ofensiva.

Dessa forma, a ferramenta desenvolvida demonstra estar em consonância com as expectativas iniciais dos alunos e também se apresenta como uma possível ferramenta para atender as necessidades de prática dos mesmos quanto à segurança ofensiva.

6. CONSIDERAÇÕES FINAIS

Segurança Ofensiva é uma filosofia de atuação dentro do domínio da Segurança da Informação. Consiste no fato do profissional aprender a pensar e atuar como o agente malicioso. Saber o que o *hacker* busca e como ele atua, e assim, compreender profundamente como uma invasão é executada, sendo capaz de executá-la, capacita o profissional a proteger mais amplamente os sistemas e infraestrutura sob sua responsabilidade.

Por necessitar de muita prática, em ambientes por vezes complexos, inacessíveis para prática ou que apresentam risco, legal ou técnico, ao praticante, pode gerar dificuldades e desencorajar aspirantes à profissionais de segurança da informação.

A fim de oferecer uma possibilidade diante desse cenário, esse trabalho apresenta uma ferramenta prática, por meio de simulação computacional, que reproduz o ambiente de um Terminal do Kali Linux, permitindo ao usuário praticar ações de segurança ofensiva, de forma didática, prática, simples e segura.

O trabalho demonstrou como um cenário pode ser criado e usado na ferramenta, e como se assemelha ao cenário real. Após a criação inicial, é de fácil reprodutibilidade e de uso viável para prática inicial, de usuários ainda pouco familiarizados com os requisitos reais. Destaca-se que apesar de apresentado um cenário específico, a ferramenta não se limita a testes de redes Wi-Fi, e evidencia a versatilidade para ser adequada para representação de quaisquer outras atividades de segurança.

O questionário aplicado reforçou a viabilidade da ferramenta, apresentando consonância entre o que oferece e o esperado pelo público-alvo. Dessa forma, representa um trabalho original com potencial de expansão e aplicação em atividades de ensino e aprendizado.

Para trabalhos futuros, os seguintes pontos são elencados:

- Inclusão de processos não lineares. Atualmente a ferramenta segue um único fluxo, com mensagens alternativas entre o resultado correto e o errado. Espera-se futuramente implementar fluxos alternativos com diferentes resultados e erros, dependendo de variações do comando executado (considerando as possíveis variáveis envolvidas);
- Possibilidade de múltiplos terminais, para tarefas que exigem essa característica;
- Execução de entrada da console (comandos do teclado), tais como CTRL + C para encerrar uma tarefa em execução, ou ↑ para recuperar o último comando digitado;
- Mecanismo de diretório, para permitir o usuário navegar em uma estrutura fictícia, definindo a localização de pseudo-arquivos com os quais esteja interagindo pelos comandos;
- Múltiplas cores para destacar instruções, mensagens de erro, resultados e comandos;
- Realizar a adequação e aplicação prática dessa ferramenta em um ambiente de ensino, para captar o *feedback* de uso a fim de otimizar o seu desenvolvimento e solidificar a proposta;

Por fim, convido o leitor para acessar a apresentação desse trabalho gravada em 20 minutos incluindo demonstração prática da ferramenta desenvolvida. Pode ser acessada a URL (YOUTUBE, 2020) ou escaneado o QR Code indicado na Figura 37.

Figura 37 – QR Code da apresentação

Fonte: Elaborado pelo autor.

REFERÊNCIAS BIBLIOGRÁFICAS

AIRCRACK-NG. **Aircrack-ng**. 2020. Disponível em: <https://www.aircrack-ng.org/>. Acesso em: 25 jun. 2020.

BANKS, J. et al. **Discrete-Event System Simulation**. 4 ed. Pearson Prentice-Hall, 2005.

BASTA, A. **Segurança de computadores e teste de invasão**. São Paulo: Cengage, 2014.

BRITANNICA. **Computer simulation**. 2020. Disponível em: <https://www.britannica.com/technology/computer-simulation>. Acesso em: 25 jun. 2020.

CISCO. **Cisco Packet Tracer**. 2020a. Disponível em: <https://www.netacad.com/courses/packet-tracer>. Acesso em: 25 jun. 2020.

CISCO. **Sistemas Operacionais de Interligação de Redes Cisco (IOS)**. 2020b. Disponível em: <https://www.cisco.com/c/pt_br/support/docs/ios-nx-os-software/ios-software-releases-110/13327-ios-early.html>. Acesso em: 25 jun. 2020.

CODECADEMY. **Learn to Code – for Free**. 2020. Disponível em: <https://www.codecademy.com/>. Acesso em: 25 jun. 2020.

CYBRARY. **Red Team vs Blue Team Review**. 2015. Disponível em: <https://www.cybrary.it/blog/2015/02/red-team-vs-blue-team-review/>. Acesso em: 25 jun. 2020.

ENGIMIND. **Microssimulação | Estação Pinheiros (Metrô São Paulo)**. 2017. Disponível em: <https://www.youtube.com/watch?v=AAFv7CJEypQ>. Acesso em: 25 jun. 2020.

EC-COUNCIL. **WHAT IS A WHITE HAT HACKER?** 2020. Disponível em: <https://blog.eccouncil.org/what-is-a-white-hat-hacker/>. Acesso em: 25 jun. 2020.

GNS3. **THE SOFTWARE THAT EMPOWERS NETWORK PROFESSIONALS**. 2020. Disponível em: <https://www.gns3.com/>. Acesso em: 25 jun. 2020.

GNS3. **Your First Cisco GNS3 Topology**. 2019. Disponível em: <https://docs.gns3.com/1d1huu6z9-wWGD_ipTSQZqy2mpaxiqzymu-YQo6at_Jg/index.html>. Acesso em: 25 jun. 2020.

HACKER SECURITY. **O que é Pentest?** 2020. Disponível em: <https://hackersec.com/o-que-e-pentest/>. Acesso em: 25 jun. 2020.

HACKER WARS. **Hacker Wars The Internet under attack**. 2019. Disponível em: <https://hackerwars.io/index.php>. Acesso em: 25 jun. 2020.

IBM. **X-Force Red**. 2020. Disponível em: <https://www.ibm.com/security/services/offensive-security-services>. Acesso em: 25 jun. 2020.

INMET. **PREVISÃO DO TEMPO**. 2020. Disponível em: <http://www.inmet.gov.br/portal/index.php?r=home/page&page=sm_previsao_tempo>. Acesso em: 25 jun. 2020.

LAW, A. M. e KELTON, D. M. **Simulation Modeling and Analysis**. 3 ed. Nova Iorque: McGraw-Hill Higher Education, 1999.

NULL BYTE. **Buy the best wireless network adapter for wi-fi hacking in 2019**. 2019. Disponível em: <https://null-byte.wonderhowto.com/how-to/buy-best-wireless-network-adapter-for-wi-fi-hacking-2019-0178550/>. Acesso em: 25 de jun. 2020.

OFFENSIVE CON. **Offensive Security Conference**. 2020. Disponível em: <https://www.offensivecon.org/>. Acesso em: 25 de jun. 2020.

OFFENSIVE SECURITY. **Information security training paths at offsec**. 2020a. Disponível em: <https://www.offensive-security.com/offsec/infosec-training-paths/>. Acesso em: 25 jun. 2020.

OFFENSIVE SECURITY. **Infosec Training and Penetration Testing**. 2020b. Disponível em: <https://www.offensive-security.com/>. Acesso em: 25 jun. 2020.

OFFENSIVE SECURITY. **Kali linux**. 2020c. Disponível em: <https://www.kali.org/>. Acesso em: 25 jun. 2020.

OFFENSIVE SECURITY. **PWK and the OSCP Certification**. 2020d. Disponível em: <https://www.offensive-security.com/pwk-oscp/>. Acesso em: 25 jun. 2020.

ORACLE. **Java**. 2020. Disponível em: < https://www.java.com/pt_BR/>. Acesso em: 25 jun. 2020.

SANABRIA, Elio. **Why the best defense is a good offensive security strategy**. 2018. Disponível em: <https://securityintelligence.com/why-the-best-defense-is-a-good-offensive-security-strategy/>. Acesso em: 25 jun. 2020.

STACK EXCHANGE. **What is the exact difference between a 'terminal', a 'shell', a 'tty' and a 'console'?**. 2010. Disponível em: <https://unix.stackexchange.com/questions/4126/what-is-the-exact-difference-between-a-terminal-a-shell-a-tty-and-a-con>. Acesso em: 25 jun. 2020.

UDEMY. **Offensive Security Engineering**. 2020. Disponível em: <https://www.udemy.com/course/offensive-security-engineering/>. Acesso em: 25 jun. 2020.

WI-FI ALLIANCE. **Discover wi-fi security**. 2020. Disponível em: <https://www.wi-fi.org/discover-wi-fi/security>. Acesso em: 25 jun. 2020.

YOUTUBE. **TCC – SI – Rafael**. 2020. Disponível em: <https://www.youtube.com/watch?v=Iyr8GoGrbYE>. Acesso em: 25 jun. 2020.

APÊNDICE A – QUESTIONÁRIO

Formulário do questionário online respondido por alunos do curso de Segurança da Informação da FATEC Americana. O questionário é composto de cinco questões.

Segurança Ofensiva para os alunos de SI da FATEC

Esse questionário visa descobrir a percepção e domínio dos alunos de Segurança da Informação da FATEC Americana em relação a segurança ofensiva. Por segurança ofensiva, entende-se práticas de pentesting, de ações de hacking (white hat) a fim de descobrir vulnerabilidades em sistemas e redes, e assim orientar suas correções.

Responda esse questionário com sinceridade. Se não souber ou não se lembrar de exemplos práticos, mencione isso.

*Obrigatório

Seu vínculo com o curso de SI da Fatec Americana é *

- ◯ formando em 2020
- ◯ formação a partir de 2021
- ◯ formado
- ◯ não sou aluno desse curso

Seu nome (não obrigatório):

Sua resposta

Você sabe realizar alguma ação se segurança ofensiva? Se sim, descreva sucintamente a(s) tarefa(s) que consegue realizar, e quais ferramentas e comandos principais você utiliza para cumprir essa(s) tarefa(s). Responda sem consultar nenhuma informação, pois é importante atestar a sua capacidade de recordar de imediato os procedimentos. Se não souber ou não lembrar, mencione isso.

Sua resposta

Considere uma ferramenta que simule um Terminal do Linux, que te permita aprender e praticar sequências de comandos para executar diferentes tarefas de segurança ofensiva, algumas por exemplo que você não conseguiria executar sem alguns recursos (p.e. wireless cracking). Uma ferramenta que te permita digitar os comandos, visualizar e compreender as saídas, e te aponte as etapas ou passos que você errou. Você acha que essa ferramenta seria útil? Você a usaria para aprender/praticar esses procedimentos?

Sua resposta

Há alguma sugestão que você possua para tornar essa ferramenta de maior interesse e utilidade para os alunos de Segurança da Informação?

Sua resposta

APÊNDICE B – RESPOSTAS DO QUESTIONÁRIO

Na relação a seguir estão apresentadas as 45 respostas obtidas após a condução do questionário do Apêndice A. Os nomes foram omitidos para preservar a privacidade dos participantes.

Questão 1	Questão 2	Questão 3	Questão 4	Questão 5
formando em 2020	M.	Não	Sim	Não
formando em 2020		Não sei	Com certeza !	Não.
formação a partir de 2021	W.	Não	Não	Não
formando em 2020		Não	sim, gostaria muito de uma ferramenta onde se pudesse praticar sem uma zona real de risco	
formando em 2020		SQLmap, Zmap e Masscan	Sim, eu acho que seria útil	Manual de comandos
formação a partir de 2021	V.	Não Sei	Sim, Seria muito útil mas acredito que demandaria muitas e constantes atualizações.	O aplicativo poderia fazer atualizações pelo menos semanais que relatassem todas as atualizações que foram feitas nos aplicativos de segurança e nos sistemas operacionais durante nesse período.
formação a partir de 2021		não	sim	
formando em 2020	E.	Sim	Sim	Nao
formando em 2020	J.	Nao lembro	Sim concerteza	Não
formando em 2020	P.	Consigo performar ações de consulta a movimentações na rede com fio e sem fio, fazendo uso de softwares como wireshark.	Sim, seria útil e seria um conhecimento novo em relação a Linux.	
formando em 2020	C.	tentativas de de brute force com ajuda do aplicativo John the Reaper junto de um arquivo de script em python.	se for semelhante as que a Cisco oferece em seus dispositivos, sim.	acho que sem uma primeira ideia nao.
formando em 2020	L.	Não sei	Muito útil e usaria com certeza	Não
formando em 2020			Sim. Usaria	
formando em 2020		Não sei.	Sim usaria	Não sei
formando em 2020			Sim	

Questão 1	Questão 2	Questão 3	Questão 4	Questão 5
formando em 2020	L.	Sim, utilizando o SQLMap do kali linux para verificar se a aplicação esta Injetável	Sim, seria e usaria a mesma	Apenas torná-la acessível para alunos e ex-alunos
formando em 2020		Não sei	Com certeza	
formando em 2020	P.	Não sei	Seria útil e usaria.	Ferramente ser didática e user-friendly
formando em 2020		Sim. Zenmap. Já mexi com outros, porém já faz um longo tempo.	Sim. Um dos maiores problemas era manter uma máquina virtual para testar essas ferramentas. Talvez uma espécie de gamificação fosse interessante. Sim.	Não ser linear como o codeacademy onde só existe uma resposta certa. Claro que no primeiras versões esse tipo defeito poderia ser ignorado.
formando em 2020	J.	Não Sei	Sim, com total certeza seria util para quem não tem tanto costume, por exemplo eu, estou mais acostumado com configurações de servidores e etc e não com segurança ofensiva.	Que a ferramenta contenha bastante explicações sobre os comandos, ou é feita uma documentação (tutorial) com todos os comandos que serão utilizados
formando em 2020	M.		Sim, essa maneira de "automatizar" processos pode ser muito util e gera agilidade	
formando em 2020	L.	segurança ofensiva, vem antes do problema realmente acontecer, alguns aplicativos nos ajudam com testes para descobrir as brechas antes de colocar no ar, alguns programas são sqlmap para arquivos de sql, ethercap como man in the midle, entre outras	Acho util a ferramenta lhe mostrar onde está o erro, assim como nos aplicativos de programação, podendo usar o programa docthor por exemplo	disponibilizar uma plataforma que rode em todos os s.i, sem uma maquina virtual

Questão 1	Questão 2	Questão 3	Questão 4	Questão 5
formando em 2020	R.	Sim, identifico vulnerabilidades e consigo monitorar através de ferramentas diversas (pentear, firewall, zabbix).	Sim	Biblioteca (significado) de siglas para ajudar a entender as saídas.
formando em 2020		não.	sim	não
formando em 2020	S.	não	sim	pentest como materia
formando em 2020	M.	Sim, aprendi bastante a parte teórica(como agir com desastres ou criar uma politica), mas o curso não me gerou muito algo prático.	Sim, acho que sim, mas acho que na prática é a melhor opção sempre	Tornar ela mais real possível
formando em 2020		Não	Não tenho nenhuma para informar no momento, seria necessario pesquisar e relembrar.	Mais utilização em sala de aula.
formando em 2020	P.	Sim. Algumas tarefas já executadas: scan de redes, scan de vulnerabilidades, sql, sniffing de redes, spoofing de rede, bruteforce de senhas, etc. Algumas ferramentas que ja usei: nmap, dig, nslookup, sqlmap, wireshark, snipper, netcat, airodump, airmon-ng, aircrack-ng, metasploit, etc.	Com certeza seria util e usaria a ferramenta para fins didáticos.	Acredito que ela deveria ser inclusa no uso dos professores durante o período letivo, se de fato ela ficar pronta e ser prática da forma que foi proposta.
formando em 2020	E.	Não lembro por hora.	Sim, desde que usada com ética e dentro da legalidade (legislação).	Webnar e Tech-Fórum são boas "ferramentas" de divulgação.
formação a partir de 2021		monitoramento da rede, nmap, pegar pacote de dados, wireshark, usar hydra no kali	sim usaria, seria algo bem util, e tudo em uma unica ferramenta sensacional	Codigo aberto e que esteja sempre disponivel
formação a partir de 2021		não sei	sim	por enquanto não

Questão 1	Questão 2	Questão 3	Questão 4	Questão 5
formação a partir de 2021		Não lembro	Sim, seria interessante essa forma de aprendizagem.	Ressaltar a importância, e manter ela de forma simples de utilizar
formação a partir de 2021		Não sei	Seria muito útil. Usaria para aprender.	Possibilidade de guardar relatórios
formação a partir de 2021	S.	Não	Sim , claro	Desmistificar o Linux. Acho que ainda assusta muita gente
formação a partir de 2021		Não	Sim!	Não
formação a partir de 2021	D.	Sim, identificar um site malicioso ou até mesmo um e-mail suspeito que possa conter vírus.	NMAP	Ter passo a passos para utilização dela, com um conteúdo fácil de entender ou até mesmo um professor que possa nos orientar de maneira clara.
formação a partir de 2021	W.	Sim, analisar trafego de rede, utilizando o khali linux, bruteforce, honeypot, ddos, rubberduccky, backdoor	Sim	
formação a partir de 2021	L.	Identificar qual ataque estamos sofrendo, parar a expansão do mesmo na rede isolando o ambiente, procurar chegar no mais próximo do início da invasão ou ataque para coletar informações, gerar relatório sobre o ocorrido, corrigir a falha de segurança, seguir com os planos de contingência da empresa.	Sim, esse método é importante para a prática de segurança.	Uma ferramenta sem muitas restrições, apenas com um "passo-a-passo" para trabalhar, um livro no caso.

Questão 1	Questão 2	Questão 3	Questão 4	Questão 5
formando em 2020	P.	Não me recordo de imediato dos procedimentos para uma segurança ofensiva, eu preciso consultar materiais pra relembrar.	Sim, concordo pois eu tenho muita dificuldade em linux, não consigo identificar falhas facilmente, uma ferramenta que auxiliasse nisso seria muito válida.	Uma ferramenta fácil de ser entendida, manuseada e aplicada, com facilidade em sanar as dúvidas encontrando-as em fóruns caso surjam as mesmas.
formando em 2020		Sim, uso de ferramentas para esse fim.	Sim	Não
formando em 2020	J.	Não lembro	Sim, com certeza!	Sua divulgação e utilização nas aulas
formação a partir de 2021	L.	não	Sim	Não
formando em 2020		Não	Seria útil, com disponibilidade de tempo eu usaria	
formando em 2020	J.	Não	Sim	Mostrar os benefícios de segurança ofensiva, e o que ela pode trazer de melhor.
formação a partir de 2021	D.	Testes de Invasão	Sim acho que seria util	Nenhuma sugestão