**Universidade Federal do ABC**

**Antonio Carlos Marchandt Zidoi**

# Análise de Vulnerabilidades em Sistemas Corporativos

**São Paulo**

**2019**

**Universidade Federal do ABC**

**Antonio Carlos Marchandt Zidoi**

# Análise de Vulnerabilidades em Sistemas Corporativos

**Monografia apresentada ao curso de Engenharia de Informação. Orientação: Prof. João Henrique Kleinschimdt**

**São Paulo**

**2019**

## RESUMO

O mundo cibernético tem trazido benefícios consideráveis na forma de comunicação que realizamos hoje. Com a velocidade de transferência de dados se tornando significantemente maior, é possível confirmar a dependência da sociedade para com esse novo meio, a Internet.

Porém, com as novas tecnologias sendo apresentadas, há um aumento na exposição de formas que podemos ser atacados por indivíduos que buscam algum ganho com essa ação. Quando grandes corporações são consideradas, esse risco se torna consideravelmente maior devido à grande quantidade de dados de usuários sendo manipulados.

Este trabalho visa demonstrar alguns ataques comumente realizados por invasores, evidenciando as formas de uso de ferramentas e tecnologia para tal. Para mitigar o risco de invasão, também são apresentadas formas de obter um nível maior de segurança para controle desses ataques apresentados.



Palavras-chave: Segurança da Informação. Internet. Redes. Teste de Invasão. Cibersegurança.

## ABSTRACT

The cyber world has brought considerable benefits in the form of communication we make today. With the speed of data transfer becoming significantly larger, it is possible to confirm the society's dependence on this new form, the Internet.

However, with the new technologies being presented, there is an increase in the exposure of forms that can be attacked by individuals who seek some gain from this action. When large corporations are considered, this risk becomes considerably higher because of the large amount of data of users being manipulated.

This work aims to demonstrate some attacks commonly performed by intruders, evidencing the ways of using tools and technology to do so. To mitigate the risk of intrusion, ways to gain a higher level of security to control these attacks are also presented.



Keywords: Information Security. Internet. Networks. Invasion Test. Cybersecurity.

## LISTA DE FIGURAS

# SUMARIO

# 1. Introdução

## 1.1 Segurança da informação

Atualmente a evolução tecnológica em sistemas de comunicação tem grande impacto nas relações interpessoais da sociedade, modificando em si a forma que as pessoas interagem em seu meio. De acordo com os levantamentos mais recentes do IBGE, atualmente dispositivos móveis estão contabilizados em cerca de 220 milhões. A partir do levantamento da mesma fonte de informação, com um total de 207,6 milhões de habitantes no país é possível concluir que há mais de um dispositivo móvel por habitante no Brasil (Meirelles,2017). Outro levantamento relevante é a disseminação da rede pelo território Brasileiro. De acordo com o último levantamento do censo, cerca de 70% das residências possuem acesso à internet.

A partir das informações mencionadas, é importante ressaltar o nível de segurança que está sendo disponibilizado a partir desse meio de comunicação, e se a proporção da procura por novas tecnologias de interconexão se adequa a proteção de seus dados pessoais. De acordo com um levantamento realizado pela empresa *Kaspersky,* considerada uma potência contra cyber ameaças pela revista *Forbes,* confirma que um a cada quatro Brasileiros já foram vítimas de um ataque chamado *Phishing*[1]*,* técnica que baseia-se na persuasão do usuário onde o mesmo é coibido a acessar alguma informação com redirecionamentos maliciosos, a partir de alguma forma que o faça ter interesse no acesso, como por exemplo sites de instituições financeiras falsas, que o leva a inserir credenciais reais que são redirecionadas para o atacante ou a falsa propaganda de prémios que são adquiridos de forma suspeita.

Com essa crescente imersão no mundo tecnológico, foi possível verificar também um crescente número de incidentes de segurança gerados a partir de deficiências de ferramentas que estão expostas para a Internet. É um fato que não existe a possibilidade de erradicar o risco de exposição quanto ao acesso à Internet, e não estar exposto a esse ambiente impossibilita também a comunicação com outros usuários e, no caso de grandes corporações o público em geral.

[1] Um em cada quatro brasileiros já caiu em phishing, golpe que rouba dados em <https://www.techtudo.com.br/noticias/2018/08/um-em-cada-quatro-brasileiros-ja-caiu-em-phishing-golpe-que-rouba-dados.ghtml>

Quando o assunto é dirigido para grandes instituições, há uma necessidade intrínseca ao negócio de técnicas e processos que visam assegurar que os dados obtidos dos usuários sejam protegidos de forma efetiva e segura, devido à grande massa de informações que são obtidas e que podem ser utilizadas para uso pessoal.

Em 2018 a empresa *Google* anunciou o desligamento de um produto utilizado por milhares de usuários devido a uma vulnerabilidade encontrada em seus sistemas que disponibilizava aos desenvolvedores o acesso de mais de 500 mil usuários e suas informações confidenciais como por exemplo nome, e-mail, ocupação, data de aniversário, entre outros[2].

Importante salientar que grande parte das invasões realizadas atualmente visam transformar as vulnerabilidades encontradas em algum valor financeiro, tanto no âmbito pessoal como o de grandes empresas. “Em 2017, os especialistas em SI notaram o aumento do interesse de criminosos cibernéticos e agentes internos pelas empresas industriais. Primeiro eles roubam os dados de usuários, planos, esquemas de processos tecnológicos, documentação técnica de engenharia e depois, monetizam essas informações” (ZUCCHERATO, 2018). Tais dados vazados podem render ganhos financeiros consideráveis para aqueles que possuem a informação, isso se dá pela possibilidade de utilização em diversas fraudes como por exemplo solicitação de cartão de crédito, envio de e-mail *spam* para injetar um *malware* na máquina da vítima, realizar engenharia social para obter vantagens do alvo, etc.

Com o aumento da exposição das corporações e da quantidade de atacantes buscando a exploração de brechas, é importante que haja investimento em tecnologias e pessoas qualificadas para conduzirem estratégias que visam a avaliação de fronteiras da rede, análise de vulnerabilidade em sistemas, realização periódica de testes de invasão na própria corporação, entre outros processos que tem como objetivo principal remediar ataques cibernéticos e evitar prejuízos consideráveis, podendo levar até mesmo na eliminação da empresa como competidor no mercado em que está alocada.

[2] Google+ is Shutting Down After a Vulnerability Exposed 500,000 Users' Data <https://thehackernews.com/2018/10/google-plus-shutdown.html>

## 1.2 Motivação

Com a crescente em tecnologias que visam o aumento de conexões e uma maior abertura para exploração de possíveis falhas, viu-se uma oportunidade de abordagem para análise de ferramentas e vulnerabilidades atreladas a elas, com finalidade de obter um maior conhecimento em testes de penetração e como os mesmos são aplicados, desenvolvidos e remediados, a fim de trazer foco a esse tema que possui atenção significativa em centros corporativos e grandes empresas de tecnologia.

## 1.3 Objetivo

O objetivo do trabalho é realizar a simulação de um ambiente corporativo, configurando e orquestrando ferramentas que normalmente são utilizadas em diversas empresas. A partir disso, encontrar possíveis vulnerabilidades relacionadas a cada ambiente evidenciando formas de ferir um dos três pilares da Segurança da Informação (Confidencialidade, Integridade e Disponibilidade).

Após as vulnerabilidades encontradas, evidenciar possíveis formas para realizar a prevenção dos ataques realizados, transformando o ambiente mais seguro utilizando ferramentas de segurança, desenvolvimento seguro, entre outros.

# 2. Fundamentação teórica

## 2.1 Confidencialidade, Integridade e Disponibilidade

A segurança da informação é regida por três grandes pilares: A confidencialidade, a integridade e a disponibilidade.

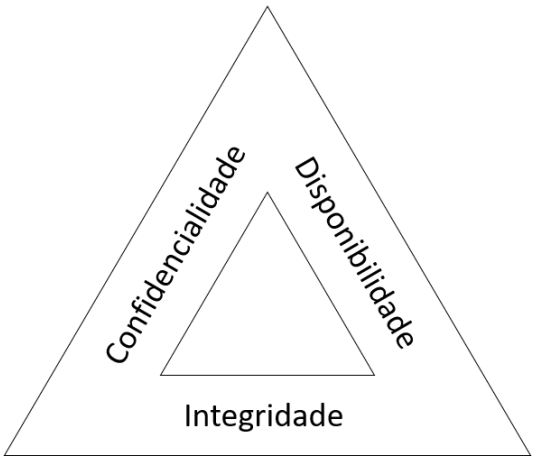


*Figura 1 - Confidencialidade, Integridade e Disponibilidade*

A confidencialidade tem como base impedir que indivíduos não autorizados tenham acesso à informação. Esse processo pode ser realizado de diversos fatores, como por exemplo o uso de criptografia, onde caso a informação seja capturada não seja descoberta. Outra forma para garantir a confidencialidade é o uso de segurança na Borda da Internet. O uso de Firewalls é um dos requisitos básicos para a proteção desse pilar. (Mike Chapple and David Seidl,2007)

A integridade assegura que não haja nenhuma modificação, sem a devida autorização, do dado que está sendo trafegado. Esse pilar em específico tem grande importância no meio da segurança da informação pois a alteração de dados em curso pode ter grandes impactos para a organização.

A disponibilidade garante que a informação esteja sempre disponível. Esse processo é importante para garantir que, caso um dado seja solicitado por um usuário autorizado independente de seu meio, essa informação esteja disponível para uso (Mike Chapple and David Seidl,2007).

Um exemplo da falta de disponibilidade é o uso de ataques *DDoS* (*Distributed Denial of Service).* A partir de um endereço da vítima, o ataque *DDoS* utiliza diversas máquinas infectadas a fim de congestionar a rede da vítima e impossibilitar que solicitações reais sejam atendidas.

Para um melhor entendimento dos efeitos de uma indisponibilidade a partir de um ataque de negação de serviço em 2015 o site GitHub, uma das maiores empresas de hospedagem de código-fonte do mundo, recebeu um ataque *DDoS* supostamente realizado por atacantes chineses devido à disponibilização de um modo de ultrapassar os *Firewalls* do país, utilizado para filtrar sites e dados administrados pelo governo. Tal ataque impossibilitou o uso da ferramenta durante dias.

## 2.2 Análise de vulnerabilidade em sistemas

A finalidade de um atacante ao procurar brechas em um determinado ambiente é ferir um ou mais dos três pilares da segurança vistos anteriormente. Seja na questão da indisponibilidade para prejudicar a corporação alvo, seja na integridade alterando informações para ganhos pessoais, ou também na confidencialidade acessando informações sem os devidos privilégios.

Os testes de penetração são comumente realizado em ambientes corporativos pela própria empresa. Isso se dá pelo fato de haver uma necessidade na busca de avaliação de como a estrutura de segurança está alocada tanto para o meio externo, como para o interno. Quando há um propósito de se realizar um teste de penetração, é comum optar pela análise de um ambiente específico (Redes de computadores, vulnerabilidade em sistemas corporativos, teste de complexidade de senhas, etc.) ou a tentativa de acesso independente do meio (equipes geralmente conhecidas como *Red Team).*

O processo realizado então pode ter três formas de ação, utilizando todo o time de conhecimento da empresa para realizar o teste, disponibilizando apenas algumas informações, ou sendo uma “caixa preta” para o *pentester*.

[3] O maior ataque DDoS já registrado teve como alvo o GitHub <https://tecnoblog.net/235518/maior-ataque-ddos-github/>

No caso da disponibilização total das informações relacionadas à corporação (*White Box*), geralmente o teste está sendo aplicado para verificar como um colaborador da própria empresa poderia agir se tivesse o desejo de atacar a empresa para fins pessoais. Os outros dois casos (*Gray Box* e *Black Box*) tem como objetivo principal avaliar a segurança do ambiente quando há uma quantidade considerável de dados disponibilizados (geralmente parceiros e empresas terceirizadas), e para usuários que possuem apenas as informações que são obtidas na rede (mais conhecido como *Google Hacking*).

Conforme a Figura 2, é possível avaliar o desenvolvimento de um teste de penetração a partir de etapas, iniciando na fase de descoberta do ambiente a ser invadido, até a finalização com a instalação de ferramentas adicionais para posteriores ataques:

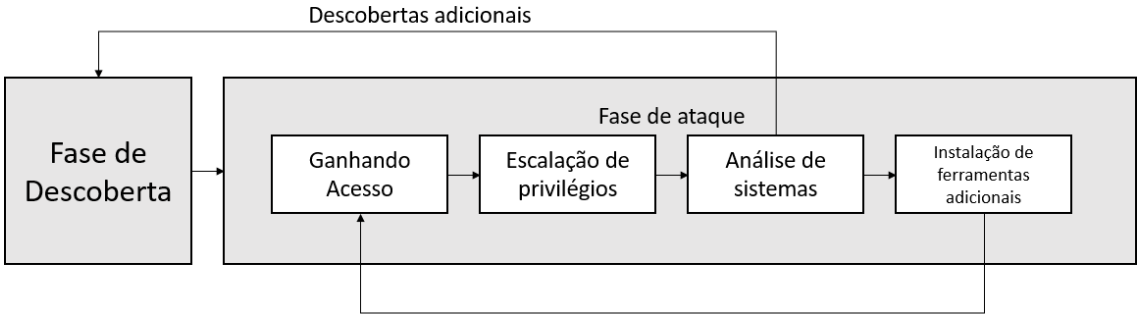


*Figura 2 - Etapas de ataque*
Fonte: Comptia CSA+ Study Guide - Mike Chapple and David Seidl

O passo inicial de um teste de Penetração em sistemas corporativos é a fase de avaliação do ambiente em suas bordas e sua segurança para o mundo externo. Essa fase é fundamental para o atacante pois assim é possível colher informações necessárias para um ataque com sucesso.

Na fase de Descoberta, os métodos utilizados podem ser Engenharia Social, reconhecimento da topologia de rede da empresa, scan de portas, identificação de serviços e suas versões, entre outros processos que visam receber mais informações do ambiente explorado.

A partir das etapas anteriores, o atacante poderá validar com maior assertividade a melhor brecha a ser explorada, seja uma versão sem *patch* de segurança, seja uma porta aberta em um determinado servidor disponível para rede, ou a própria equipe que compõe a empresa e não possui requisitos mínimos de segurança e poderiam ser exploradas. Quando um atacante consegue com sucesso entrar na rede corporativa da empresa, sua função a partir desse momento é descobrir formas de se tornar um administrador *root* para que suas permissões possibilitem conquistar seus objetivos.

Na etapa pesquisa de sistemas, o atacante tem como principal objetivo validar os demais sistemas do ambiente para elevar seus privilégios, realizando ações laterais (Infectando outras máquinas) ou procurando brechas na ferramenta que ele já conquistou.

Para finalizar, após ter realizado com sucesso o ataque é fundamental que todos os registros de ações que foram realizadas sejam deletados, e que um *Backdoor* (criação de uma “porta dos fundos” para acesso do invasor) seja criado para facilitar novas invasões (Mike Chapple and David Seidl,2007).

Realizando a engenharia reversa do que foi dito anteriormente, para que os ataquem sejam evitados é importante que seja analisado o processo, validando todos os passos e evitando que eles sejam realizados. Para o primeiro momento, ao se realizar uma busca de informações a empresa deverá tomar algumas precauções para que a pesquisa de dados seja efetiva.

Diversas empresas possuem ferramentas que fazem com que os *scans* realizados em seus servidores sejam identificados e bloqueados. Essa função elimina a possibilidade de haver um atacante descobrindo informações por qualquer tipo de *scanner* que possa estar sendo utilizado.

Ao se ganhar acesso, uma possibilidade de coibir as ações laterais é eliminar a comunicação entre máquinas/servidores dentro da instituição por meios incomuns, bem como segregar as redes mais críticas. Dessa forma haverá uma possibilidade menor de um incidente em dados críticos quando uma máquina é infectada.

O uso de duplo fator de autenticação em acessos privilegiados também é comum em grandes empresas. Um dos requisitos da certificação *PCI DSS (Payment Card Industry Data Security Standards),* certificação mundial para empresas que utilizam processos de pagamento via cartão, é o uso de MFA (*Multi-Factor Authentication)* para o acesso ao ambiente de forma remota (Todd Bey, 2018).

## 3. Metodologia

Para o processo de implementação do trabalho, algumas ferramentas foram selecionadas para a realização de uma simulação da infraestrutura de um ambiente corporativo.

Em grandes corporações é comum o uso de grandes Datacenters para o armazenamento de dados, e esses dados geralmente são inseridos em bancos de dados para facilitar a manutenção, consulta e análise. No projeto foi utilizada a ferramenta *MySQL* para essa simulação onde um banco de dados foi construído em uma máquina *Linux* Versão *Ubuntu 16.04*.

Além disso, também ocorre o processo de compra de sistemas de outras empresas para uso no ambiente corporativo, dessa forma essas ferramentas são incorporadas para o uso em diversas áreas que irão se beneficiar do desenvolvimento realizado em outras corporações e obtido para uso interno. No caso em questão, foi realizado o uso de uma ferramenta proprietária de acesso e gestão de câmeras de segurança. Na simulação, foi utilizado o modelo DVR-08 da empresa Elgin.

Para finalizar, é importante que haja um ambiente de rede para os colaboradores e usuários poderem utilizar a rede para comunicação. Com isso é possível tanto realizar o acesso à Internet, como restringir o uso da rede local apenas para usuários autorizados. Para a simulação da rede, foi utilizado um roteador *LinkSys E900.*

Para o ataque, uma máquina com o sistema operacional *Kali Linux* foi utilizada. Esse sistema tem como função principal disponibilizar ferramentas para um processo de análise de vulnerabilidades, bem como a exploração dessas falhas previamente encontradas. O Kali Linux é predominantemente utilizado por corporações e Hackers que buscam realizar uma análise crítica do sistema que está sendo verificado.

Dessa forma, o ambiente estudado foi construído conforme a Figura 3. É possível verificar máquinas de usuários realizando o acesso à rede local, e duas ferramentas também conectadas à rede via conexão *Wi-Fi.* Com o ambiente parametrizado, ataques são realizados no ambiente a partir das vulnerabilidades encontradas nos sistemas utilizados (Protocolos vulneráveis na rede local, possibilidade de indisponibilidade no serviço de Banco de Dados e análise de vulnerabilidade em uma aplicação de gestão de câmeras de segurança).

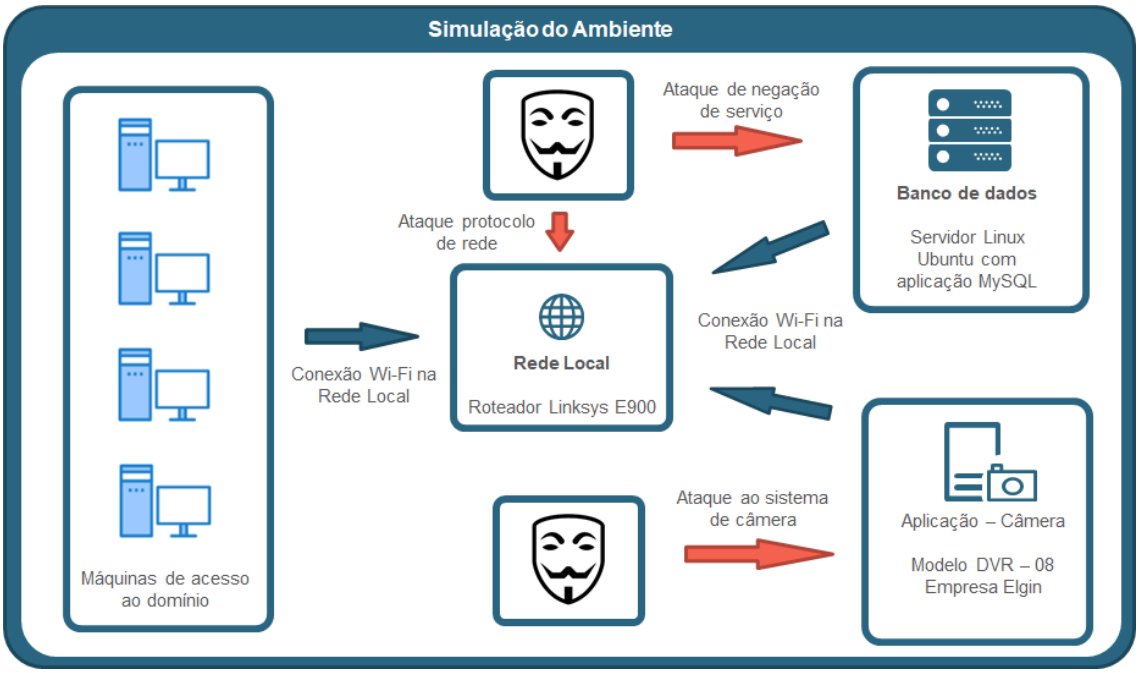


*Figura 3 - Ambiente Simulado*

A partir das parametrizações das ferramentas, é possível obter um vetor de ataque para cada ambiente e, após isso desenvolver técnicas para aumentar o nível de segurança da simulação realizada.

Para o ataque aos ambientes foram realizados ataques de negação de serviço, avaliação dos protocolos de segurança de redes para o roteador e avaliação de Vulnerabilidades com o *Nessus* para a ferramenta escolhida. Para a Defesa, foram disponibilizados protocolos seguros e autenticação via *Radius,* alteração de regra de *Firewall* para bloquear ataques *DoS* e criação de um servidor *Front-end* para mitigar riscos da aplicação proprietária.

Dessa forma, o modelo de aplicação do ambiente, o desenvolvimento de ataque e efetivação de defesa pode ser avaliado conforme a Figura 4. É possivel analisar a implementação de três aplicações distintas (evidenciadas na chave “APLICAÇÕES”), as vulnerabilidades que foram exploradas a partir da parametrização desses sistemas (evidenciadas na chave “ATAQUES”) e os modelos de defesa propostos para cada uma da falhas encontradas (evidenciadas na chave “DEFESA”), mitigando o risco de novas invasões.

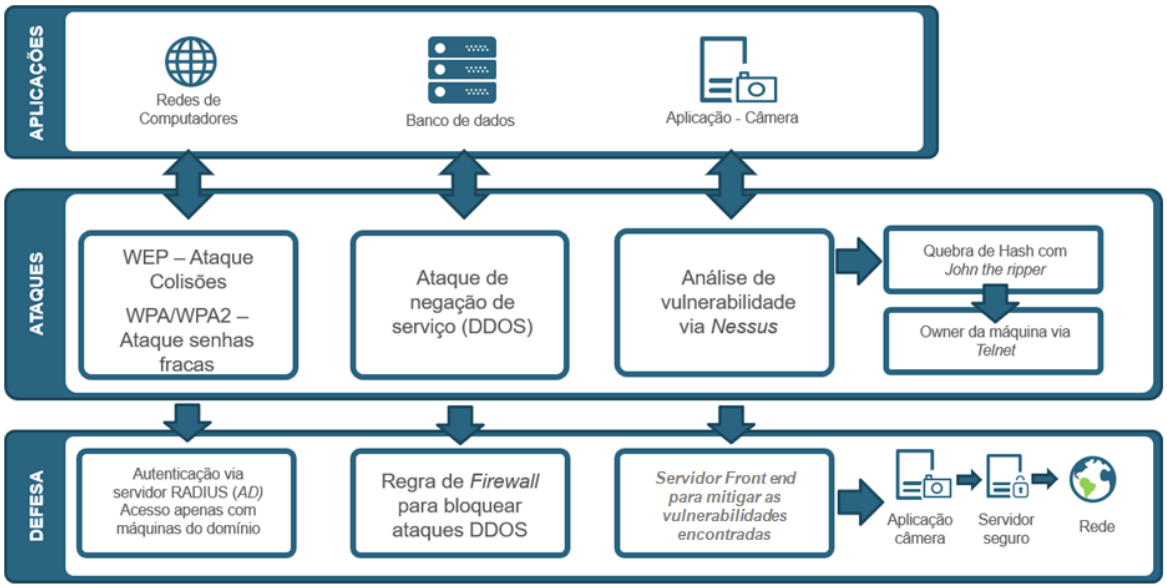


*Figura 4 - Mapping de ataque*

## 3.1 Ameaças

### 3.1.1 Ataque DoS

Ao se desenvolver um Banco de dados, quando este se comunica com a organização existem algumas precauções que devem ser tomadas, como por exemplo sua comunicação, suas configurações, entre outros aspectos para que os três pilares sejam assegurados. Para corromper um dos pilares (disponibilidade), foi utilizada a seguinte ferramenta:

O *LOIC* (Canhão de Ions de Órbita Baixa) é uma ferramenta de código aberto bastante conhecida no mundo *hacker* por sua história junto aos *Anonymous*. Uma

máquina solitária normalmente não possui poder de tráfego o suficiente para realizar um ataque de negação de serviço em servidores, porém quando combinados com milhares de outros computadores, o prejuízo para a vítima pode ser considerável.

Um dos primeiros ataques coordenados dos hackers intitulados como *Anonymous* foi nos servidores da igreja Cientologia. O software foi disponibilizado em um dos fóruns utilizados pelo grupo e foi baixado milhares de vezes (Parmy Olson, 2014).

A partir disso, houve um ataque em massa de *máquinas zumbis* infectadas com um malware para realizar o ataque, e até mesmo pessoas que entraram no fluxo de atacantes. O intuito dessa ferramenta é, a partir de um IP fixo realizar o envio de dados (sem o intuito de realmente realizar a conexão com o servidor) com finalidade de impactar a rede para que as conexões que realmente são verdadeiras vejam uma rede completamente congestionada e sem a possibilidade de conexão.

Conforme a Figura 5, a aplicação *LOIC* possui alguns parâmetros que precisam ser inseridos para o correto funcionamento da ferramenta. Na área *Select your target,* o alvo desejado deve ser inserido com uma URL ou um IP. Após isso as informações de ataque devem ser inseridas:

- *HTTP Subsite* – Caso haja um subsite dentro do site principal, o mesmo deve ser inserido nesse campo.
- *Port* – Inserir a porta alvo (No caso em questão, o serviço SQL está disposto por padrão na por 3306).
- *Method* – Inserir se o protocolo de ataque será TCP ou UDP.
- Threads – Inserir a quantidade de conexões que o *LOIC* tentará estabelecer.

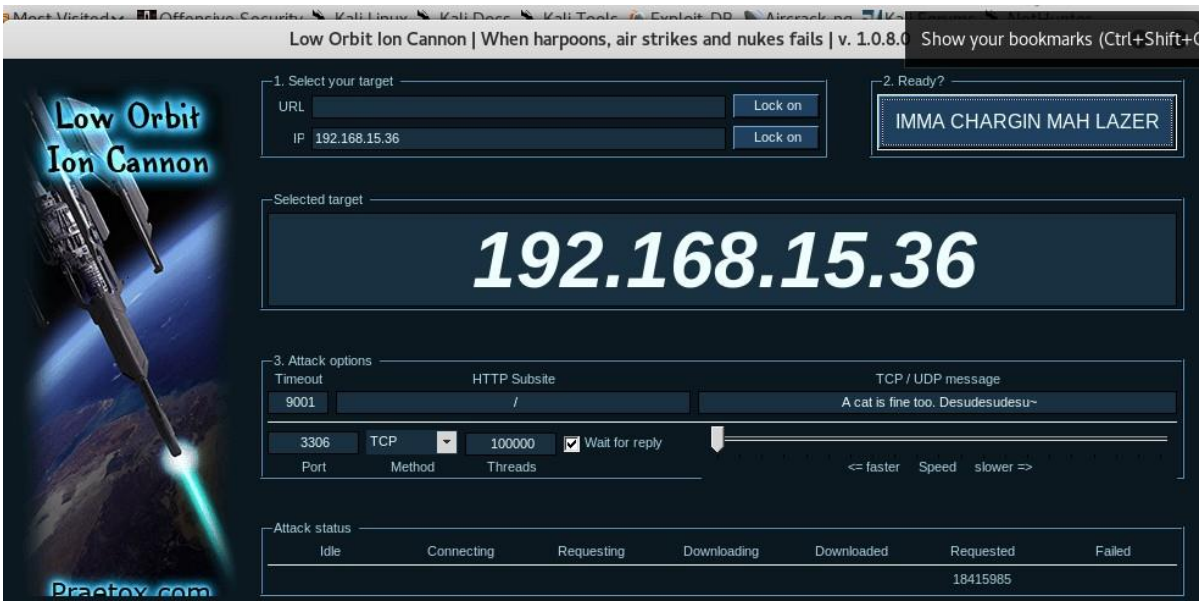


*Figura 5 - Aplicação LOIC*

### 3.1.2 Superfície de ataque

Em ambientes corporativos, é comum a compra de ferramentas e equipamentos de outras empresas para uso próprio. Isso se dá pelo fato de não haver a necessidade da criação de todas as ferramentas que serão utilizadas pela corporação, simplesmente por outras empresas serem especializadas em determinadas áreas e possuírem sistemas que são significantemente melhores ao que poderiam ser desenvolvidos internamente.

Porém, um cuidado deve ser levado em consideração quando se toma uma ação de compra de outras ferramentas, o quanto se está seguro na agregação de um novo sistema de outro fornecedor.

Deve-se levar em consideração se o sistema possui os requisitos mínimos de segurança e que não existem brechas que podem ser exploradas por atacantes maliciosos. Essas brechas podem ser descobertas por diversas ferramentas existentes no mercado que tem a função de mapear um determinado ambiente a fim de descobrir possíveis vulnerabilidades que podem ser exploradas.

A partir do *scan* de portas e vulnerabilidades, é possível verificar a possibilidade de investida com o que se é apresentado. Com as falhas que serão demonstradas, é possível encontrar formas de acesso e elevação de privilégios.

Existem diversas ferramentas de mercado que são utilizadas para Scanners, como por exemplo o Nikto, Nexpose ou Qualys, porém focaremos no *scan* de vulnerabilidades *Nessus*.

O *Nessus* possui uma interface que segrega por escopo de atuação para com a avaliação de vulnerabilidades que se deseja pesquisar. De avaliação de credenciais a análise de vulnerabilidades de redes os quais podem ser analisados na Figura 6, o *scan* tem a possibilidade de fazer uma avaliação das possíveis brechas de segurança que deverão ser exploradas.

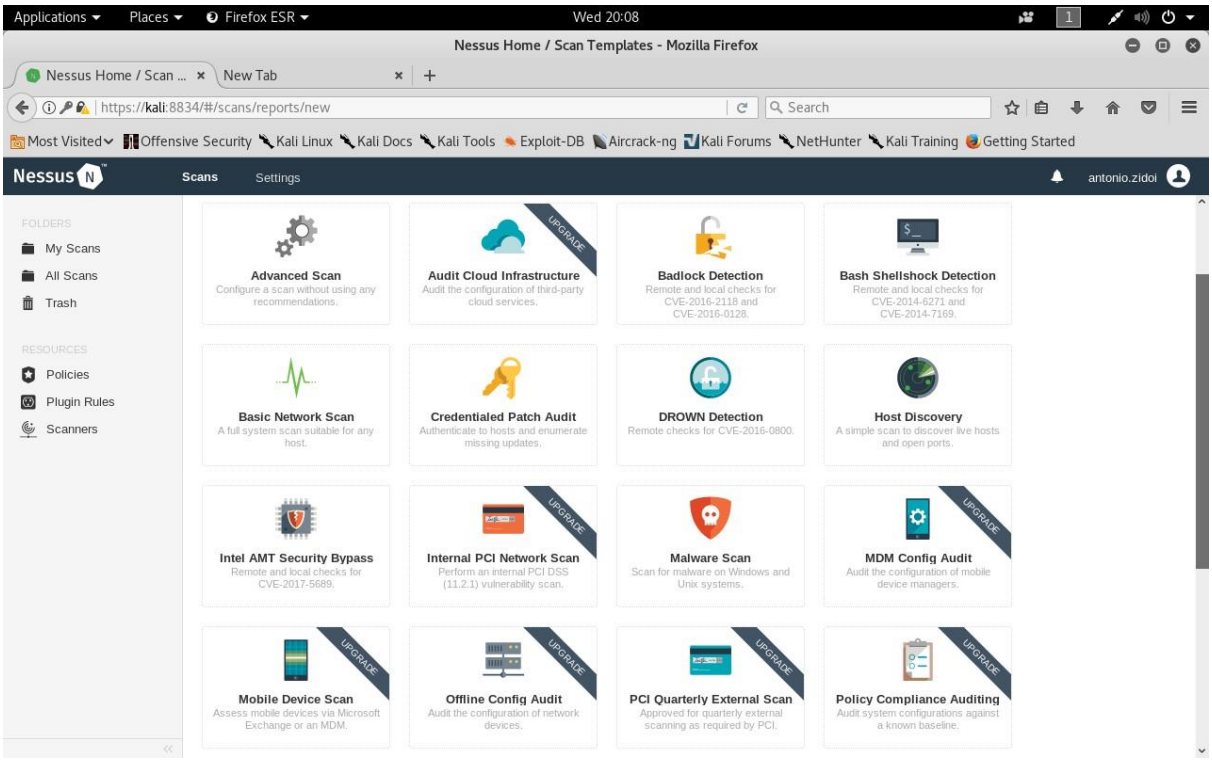


*Figura 6 - Aplicação Nessus*

O Sistema possui um modelo de parametrização de ameaças chamado CVSS (*Common Vulnerability Scoring System*), um modelo que utiliza alguns métodos como avaliação da necessidade de estar fisicamente alocado para concluir o ataque ou a quantidade de níveis de autenticação para a exploração para avaliar o quanto uma determinada falha é Crítica, Moderada ou Baixa.

Esse modelo de avaliação é utilizado para analisar as principais prioridades que o especialista deve focar e avaliar para sanar as falhas encontradas.

A ferramenta utilizada para *scan* de portas é o Nmap. O interessante da ferramenta é o poder de disponibilização de informações que ela possui. Ao apontar para um IP específico, podemos determinar algumas informações cruciais para uma análise de vulnerabilidade.

Uma das informações mais importantes da ferramenta é a possibilidade de verificação de versão da ferramenta que está utilizando a determinada porta. Com essa informação, podemos verificar se existe alguma possível vulnerabilidade nas ferramentas que estão sendo utilizadas, as possíveis vulnerabilidades de sistemas sem atualizações, entre outros.

### 3.1.3 Ataque protocolo de redes

O protocolo *WEP* foi criado para oferecer segurança para redes sem fio, porém a tecnologia foi rapidamente quebrada por suas falhas de segurança. O protocolo funciona com uma chave dividida em duas partes: o IV (*Initilization Vector*) e a chave de criptografia.

O processo de codificação de uma mensagem é a realização de uma combinação XOR do texto simples e a chave, gerando em sí o texto criptografado conforme a Figura 7:

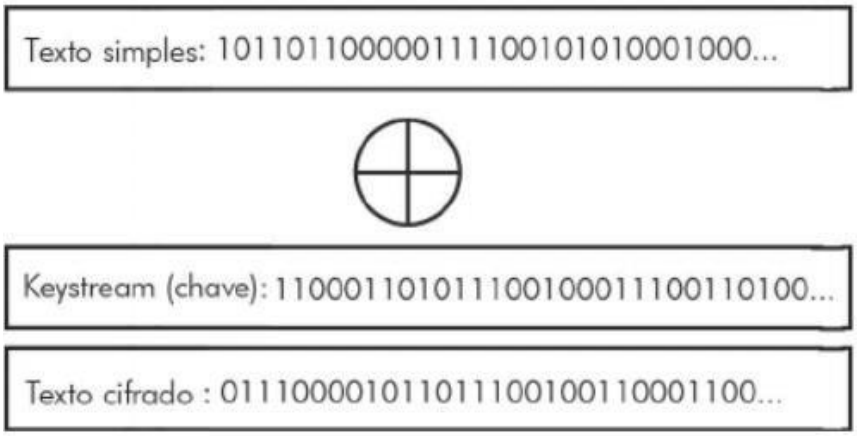


*Figura 7 - Colisões protocolo WEP*

Fonte: Comptia CSA+ Study Guide - Mike Chapple and David Seidl

O IV é uma chave que é adicionada para gerar aleatoriedade na criptografia, porém o problema com o processo é o tamanho de sua codificação. Com um total de 24 bits, a ideia original de se realizar um processo aleatório não funciona de forma efetiva, pois há um total de 2^24 (16.777.216) valores.

O processo de captura da chave criptográfica consiste em realizar a monitoração de comunicação de pacotes, obtendo informações o suficiente para realizar o processo de colisão, conseguindo enfim ter posse do texto claro.

Havendo pacotes com o mesmo IV, como a chave de comunicação é sempre a mesma pode-se realizar o processo inverso e obter a chave, texto claro e a forma de acessarmos a rede a partir de então.

Com as vulnerabilidades do protocolo *WEP,* houve a necessidade da criação de um novo protocolo para atendimento dos sistemas que ficavam cada vez mais robustos e necessitavam de um processo mais seguro para comunicação.

Dessa forma, o protocolo *WPA* foi criado. também conhecido como *TKIP (Temporal Key Integrity Protocol),* essa nova tecnologia veio para suprir os pontos fracos deixados pelo seu antecessor. Agora com uma chave de 148 bits, a nova tecnologia garante a efetividade de sua segurança com uma chave de criptografia única para cada comunicação (3,56^44 valores diferentes).

A efetividade da invasão de um ataque às redes WPA/WPA2 está na senha utilizada pelo administrador. Ao se capturar o tráfego de início (*HandShake)* é possível comparar os *hashes* utilizados na autenticação com um dicionário pré-configurado.

A ferramenta que será utilizada é conhecida na avaliação e detecção de redes vulneráveis, e também é disponibilizada pelo sistema operacional Kali Linux. O *Aircrack-ng* é um sniffer que tem como função principal realizar a detecção de redes, bem como realizar a quebra de protocolos mais vulneráveis por força-bruta. Ele será utilizado para a verificação da proteção da rede, e como o processo de identificação de senha será realizado.

Importante que, quanto mais fraca uma senha (Não preenchendo os requisitos mínimos de segurança: Letras/Números/Caracteres Especiais), mais fácil é realizar o processo de quebra de senha com força bruta.

**3.2 Controle de Segurança (Defesa)**

Para haver um controle efetivo de Segurança no ambiente que está sendo estudado, é importante que processos e ferramentas sejam implementados para garantir uma maior efetividade e assertividade na mitigação de possíveis brechas que poderiam ser exploradas caso não houvesse uma análise crítica a esse tema. Conforme descrito abaixo, diversas ferramentas são necessárias para haver um controle de segurança do que está sendo monitorado e testado. Com a implementação desses sistemas, é possível verificar um maior nível de maturidade para com o ambiente amostrado.

### 3.2.1 Firewall

O Firewall é um dispositivo de rede que tem como principal função ser o primeiro componente a fazer face com a rede externa. Como o próprio nome, o conceito da ferramenta é ser a “parede corta-fogo” do ambiente seguro e impedir o contato não monitorado da rede insegura (Internet). Essa ferramenta evoluiu consideravelmente no decorrer dos anos. Iniciando como um analisador de protocolos e portas, atualmente o Firewall tem tecnologia para avaliar até mesmo o estado da sessão que está sendo realizada, se já está estabelecida, se está tendente a ser fechada, etc.

No caso de Firewall local, o seu conceito é semelhante pois a proteção está focada em impedir que o ambiente inseguro se comunique com o seguro, porém nesse caso a ferramenta analisa o comportamento e interação com o host local e suas conexões com outros dispositivos até mesmo se estiverem em uma mesma rede.

Uma das diversas possibilidades em seu uso é a possibilidade de realizar a filtragem de pacotes a partir de diversas regras que podem ser implementadas, dessa forma é possível obter restrições de acordo com a necessidade de quem está implementando a ferramenta.

### 3.2.2 Active Directory

O AD é uma ferramenta da Microsoft que tem como função principal realizar o gerenciamento e administração de forma centralizada de dispositivos e usuários, que são descritos como objetos. Essa ferramenta permite que sejam criados grupos de permissionamento para objetos cadastros agindo como uma base de autenticação e autorização de ferramentas.

A partir de agrupamentos (Domínio, Unidades Organizacionais, grupos, etc.) é possível gerar processos que impactarão todos os objetos atrelados a ele. Um exemplo é aumentar o nível de criticidade de senha de usuários cadastros em uma determinada OU (Unidade Organizacional).

Identificando a diretrizes corretas, é possível realizar de forma otimizada e simples a mudança de parâmetros de todos os usuários cadastrados em apenas um processo, ao invés de realizar de forma manual para todos os usuários da rede.

Utilizando o protocolo Radius, com o AD é possível obter o conceito de AAA (Autenticação, Autorização e Auditoria), que visa construir um modelo que gere de forma centralizada a validação de credenciais que serão utilizadas no acesso. Com a rede implementada, processos onde apenas dispositivos registrados nesses agrupamentos utilizem a rede corporativa podem ser configurados. Dessa forma o roteador utilizado se comunicará com o AD para a validação se o dispositivo e/ou o usuário que estão solicitando acesso tem a permissão para isso.

A partir disso é possível realizar a validação de senha, bem como estipular o nível de complexidade que cada usuário terá, eliminando a forma “doméstica” de autorização de senhas, com uma senha padrão para todo o acesso que será realizado.

### 3.2.3 Hash

O *Hash* é um algoritmo criptográfico *one-way function* isso é, a partir de diversas operações realizadas com a mensagem que está sendo enviada, “não é possível” realizar a operação de volta para descobrir qual dado de fato foi enviado pelo destinatário.

Um exemplo de uso do Hash é no armazenamento de senhas. Para que não haja o armazenamento de senha do usuário em texto claro, é fundamental que essa informação seja registrada apenas em seu Hash. A partir das credenciais do usuário, essa senha é transformada em Hash e validada com a informação armazenada. Caso essa comparação esteja correta, há sucesso na autenticação do usuário.

# 4. Simulação de ambiente corporativo

## 4.1 Implementação

### 4.1.1 Banco de Dados

Os testes e experimentos realizados em Banco de dados apresentados aqui podem ser realizados via teste externo (Fora da rede local) como interno. Isso se dá pois independente da conexão do atacante e a base de dados, haverá um IP de destino e uma porta de comunicação

Para a criação do Banco de dados, foi utilizada uma base teste onde a informação em si não é relevante, porém o que está sendo testado é a indisponibilidade da ferramenta e possíveis forma de contorná-la.

Na Figura 8 pode-se verificar o banco de dados via aplicação *MySQL* em funcionamento a partir de um servidor com IP fixo 192.168.1.124. Com as configurações parametrizadas, é possível realizar o acesso a partir de outros dispositivos na mesma rede:

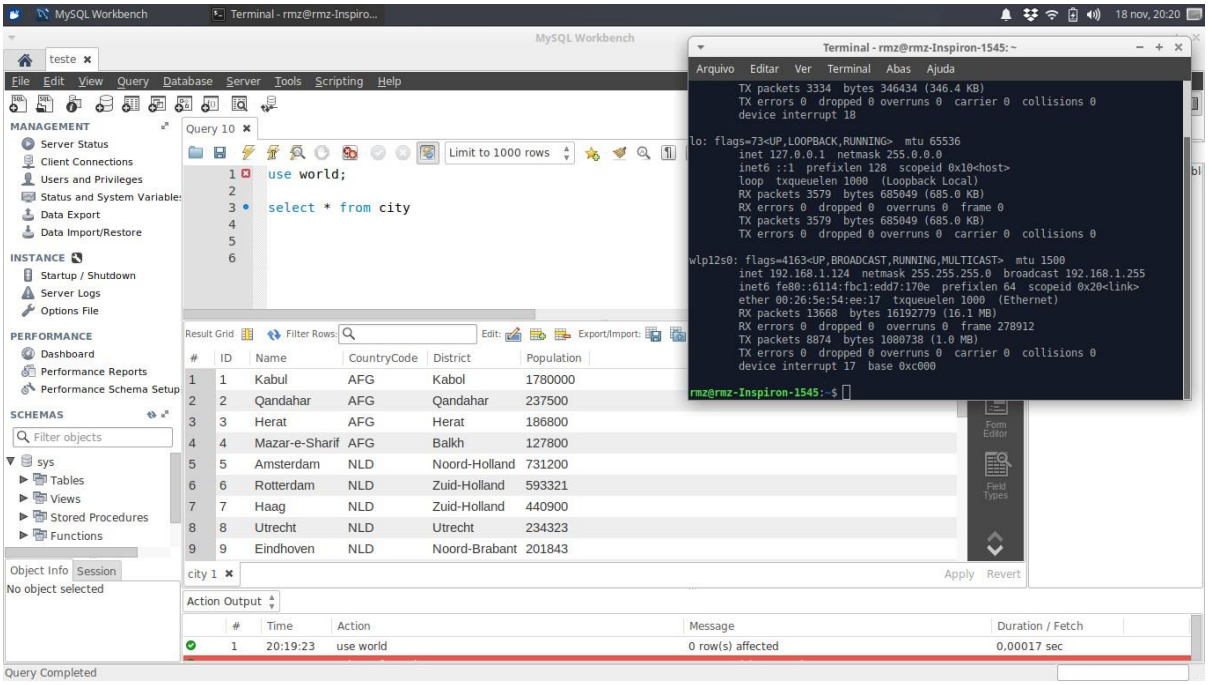


*Figura 8 - Aplicação MySQL Workbenck / IP de acesso ao servidor*

Para que houvesse a comunicação entre os dispositivos e outras máquinas, foi utilizado o UFW. UFW (*Uncomplicated Firewall)* é um Front da tabela IPTABLES. A partir dele, há a possibilidade de realizar a abertura de regras de firewall de IPs específicos, e portas selecionadas.

A liberação foi realizada para que fosse disponibilizada a opção de entrada do banco a partir de uma conexão remota, dessa forma também houve a criação de uma vulnerabilidade de acesso para um possível ataque de negação de serviços (DDoS). A comunicação com a aplicação é feita via porta 3306 conforme evidenciado na Figura 9.

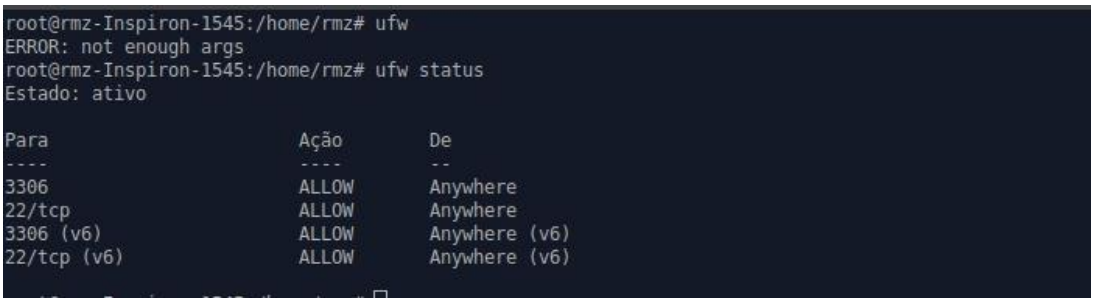


Figura 9 - Aplicação UFW

Para que a comunicação entre máquinas em uma rede local fosse testada, um outro dispositivo foi necessário. No lado do cliente, o SQL *WorkBench* foi instalado e para que houvesse o acesso ao servidor (parametrizado anteriormente), foi cadastrado como *Hostname* o IP de destino, que no caso do teste é o 192.168.1.124. Realizando um teste de conexão, na Figura 10 pode-se verificar com sucesso uma conexão à base de dados:

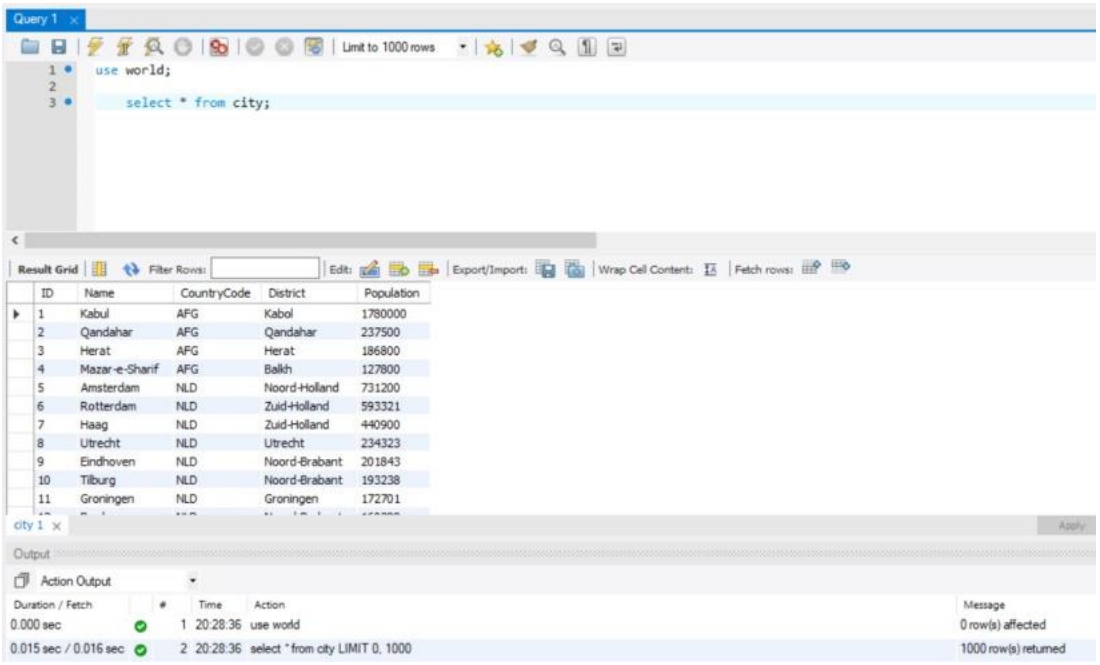


*Figura 10 - Conexão entre cliente e Servidor*

Importante notar o tempo de duração da pesquisa e de busca (0,015 seg / 0,016 seg), pois esses dados serão utilizados para uma futura avaliação do ambiente.

### 4.1.2 Ferramenta proprietária de acesso à câmeras de segurança

O Sistema foi configurado para estabelecer uma conexão *HTTP* para o ambiente local onde a ferramenta está alocada, dessa forma é possível realizar o acesso das informações disponibilizadas pelo sistema de câmeras através de uma página via *Internet Explorer*. Na Figura 11., é demonstrada a forma que as informações de conexão foram parametrizadas, onde o endereço IP e a porta foram configurados de forma à disponibilizar o acesso *HTTP* e o *Gateway* e DNS primário configurados para direcionar o tráfego para o roteador possibilitando acesso das informações disponibilizadas para a rede local.

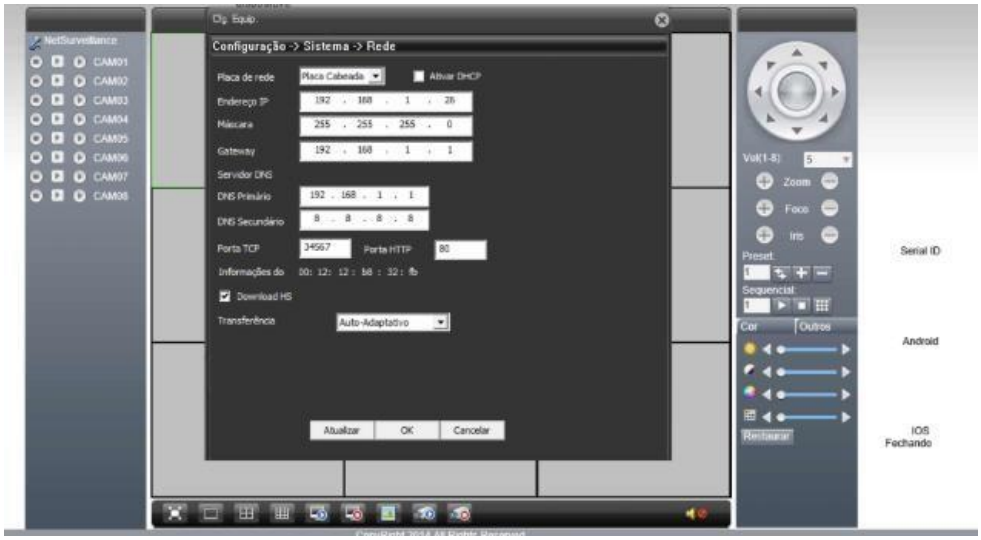


*Figura 11 - Configuração de Serviço - Aplicação Proprietária*

Com isso, foi possível realizar o acesso via Internet Explorer a partir do IP http://192.168.1.26:80. Com a liberação de acesso para a rede, a avaliação das possíveis brechas de segurança da ferramenta e como elas serão exploradas podem ser realizadas.

## 4.2 Banco de Dados

### 4.2.1 Ataque

Para realização do ataque ao banco de dados um novo dispositivo foi inserido à rede com a aplicação LOIC ativa, dessa forma foi possível obter uma comunicação com sucesso desse novo dispositivo com o servidor onde o banco de dados estava alocado, estando vulnerável à um possível ataque de negação de serviço.

Para que o ataque tivesse sucesso, foram necessárias 6 instâncias diferentes. A cada nova aplicação aberta e parametrizada, um maior atraso na pesquisa ao banco de dados ocorria. Na Figura 12 pode ser verificado o IP de destino o qual o ataque foi direcionado:

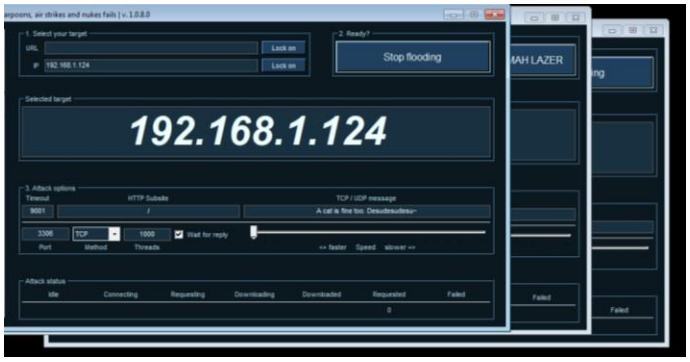


*Figura 12 - Parametrização LOIC*

Com as 6 aplicações em conjunto, pode-se observar uma imobilização total da consulta na ferramenta. A Figura 13 representa uma tentativa de acesso do Cliente (Comunicação real) com o servidor que possui o MySQL exposto para a rede. É possível verificar que após algumas tentativas há uma indisponibilidade total da ferramenta para um acesso legítimo:

| | # | Time | Action | Message |
|---|---|---|---|---|
| | 12 | 21:00:48 | use world | 0 row(s) affected |
| c | 13 | 21:00:49 | select * from city LIMIT 0, 1000 | 1000 row(s) returned |
| | 14 | 21:03:03 | use world | 0 row(s) affected |
| c | 15 | 21:03:03 | select * from city LIMIT 0, 1000 | 1000 row(s) returned |
| | 16 | 21:04:04 | use world | 0 row(s) affected |
| | 17 | 21:04:30 | select * from city LIMIT 0, 1000 | Error Code: 2013. Lost |

*Figura 13 - Indisponibilidade de Servidor durante ataque*

Na figura 14 é relacionado o tempo de comunicação de um usuário legítimo (referenciando à Figura 13) validando o tempo de comunicação que exigido a partir da quantidade de aplicações realizando o ataque simultaneamente. A partir de 5 aplicações verifica-se que o usuário realizando o teste via console MySQL recebe um erro de comunicação com o servidor pelo tráfego estar congestionado, tornando a informação indisponível.

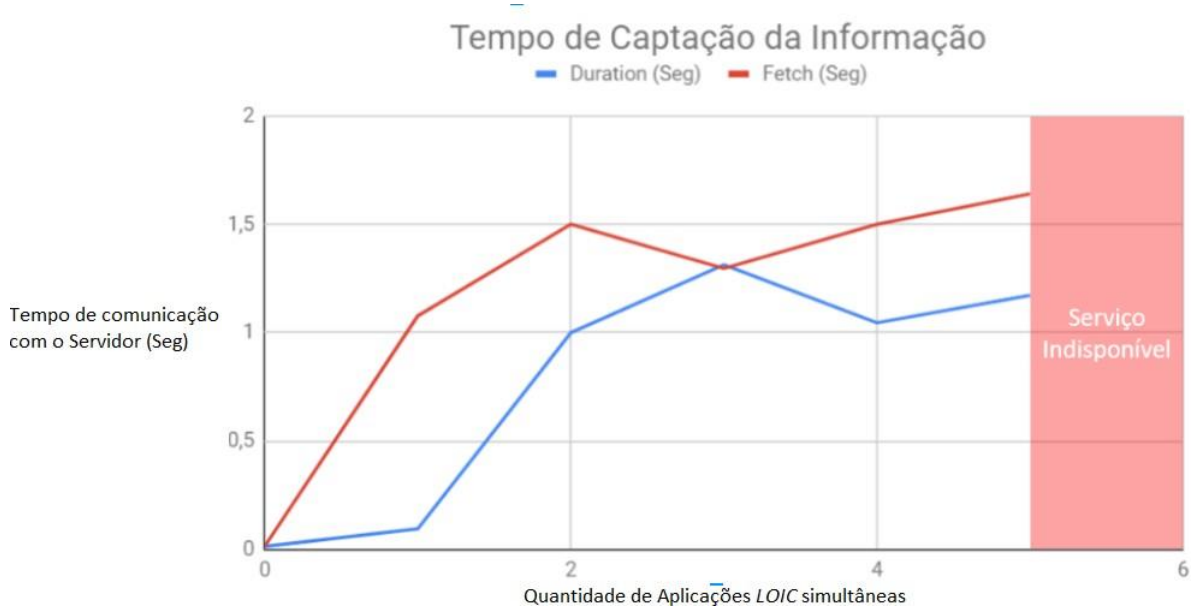


*Figura 14 - Pontos de comunicação X Aplicações LOIC*

*Fetch Time* - Relação de tempo para a busca de informação em um servidor ser realizada. Essa busca reflete o tempo de conexão entre os dispositivos.
*Duration Time* - Tempo de execução da *Query*.

Dessa forma, é possível observar o ambiente simulado da seguinte forma na Figura 15:


*Figura 15 - Estrutura de simulação 1*

### 4.2.2 Defesa

Para solução da indisponibilidade gerada pelo ataque DoS, algumas regras deveriam ser avaliadas e inseridas no servidor para que o serviço continuasse ativo e disponível para consumo

Para isso foi preciso implementar uma regra de firewall via *IPTABLES* que visava a identificação de conexões com a plataforma a partir de uma determinada porta. Com as conexões que estavam sendo solicitadas ao servidor, o *script* realizava o monitoramento do tráfego a fim de avaliar a quantidade de conexões por IP (que no caso parametrizado, eram limitadas a dez conexões) realizando o *DROP* após o limite, e estabelecia um valor máximo para a quantidade de usuários conectados ao servidor (no caso, 150 conexões).

Dessa forma, a seguinte regra foi inserida:

```
iptables -t filter -I INPUT -p tcp --dport 3306 -j ACCEPT
iptables -t filter -I INPUT -p tcp --dport 3306 -m state \
  --state RELATED,ESTABLISHED -j ACCEPT
iptables -t filter -I INPUT -p tcp --syn --dport 3306 -m connlimit \
  --connlimit-above 10 --connlimit-mask 32 -j DROP
iptables -t filter -I INPUT -p tcp --syn --dport 3306 -m connlimit \
  --connlimit-above 150 -j DROP
```

O intuito da implementação foi limitar a quantidade de conexões do servidor, bem como limitar a quantidade de conexões um mesmo IP poderia ter.

A partir da implementação da regra, uma diminuição expressiva no tempo de comunicação entre a máquina cliente e servidor foi observada. Essa observação confirma a diminuição no congestionamento de conexões com o servidor.

Podemos avaliar a efetividade da ação a partir do tempo 01:47:00 na Figura 16, onde verificamos uma diminuição considerável no tempo de resposta do servidor com uma solicitação de pesquisa do cliente (Ainda com o ataque em andamento).

Ao realizar a retirada dos parâmetros realizados anteriormente, verifica-se a diminuição do tempo de duração de resposta relativa ao primeiro momento, onde não havia nenhuma técnica de defesa confirmando a efetividade da ação:

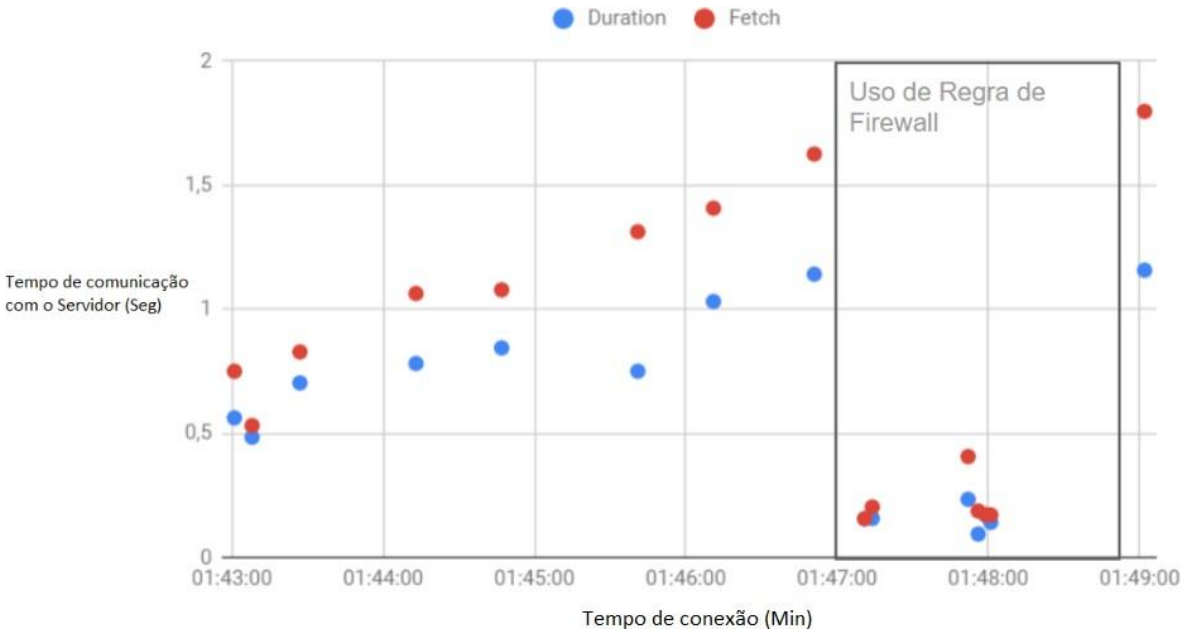


*Figura 16 - Avaliação de comunicação com servidor*

O processo amostrado relaciona apenas um IP realizando um ataque de negação de serviço em um servidor em ambiente produtivo, porém quando ataques de negação de serviço são realizados geralmente milhares de dispositivos são utilizados para realizar o ataque em um único ambiente.

É possível afirmar que não existe uma solução definitiva para a mitigação do risco de ataques de negação de serviço, visto que é necessário que uma plataforma de serviços esteja disponível para a Internet, e conexões de diversos usuários sendo realizadas simultaneamente. O que geralmente é feito para minimizar essa indisponibilidade é a tentativa de identificação de usuários que não possuem o intuito de realizar uma conexão legítima, investimento e infraestrutura para balancear as conexões que são realizadas para que servidores estejam sempre disponíveis, e focar em ambientes paralelos onde diversos servidores são utilizados para um mesmo propósito (ato de *Clusterização).*

Dessa forma, é possível observar o ambiente simulado da seguinte forma na Figura 17:

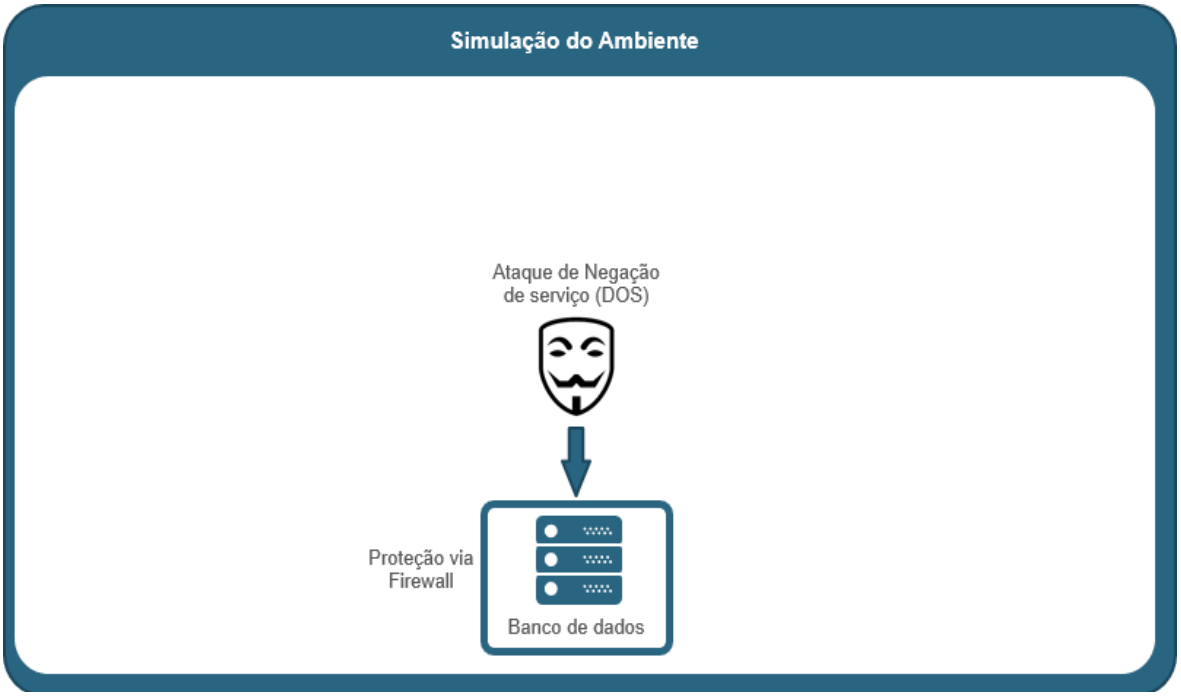


*Figura 17 - Estrutura de simulação 2*

## 4.3 Ferramenta proprietária de acesso à câmeras de segurança

### 4.3.1 Ataque

Após o ataque de negação de serviço, um novo ataque foi realizado. A partir da ferramenta de câmeras previamente inserida no ambiente, pode-se analisar possíveis vulnerabilidades.

Como primeira avaliação da ferramenta, ao ter posse do manual verificamos que está disponível para o usuário acessar o sistema com usuário default e sem senha registrada. Esse padrão para as ferramentas no geral são comuns e abrem brechas significativas para o usuário. Tal forma é escolhida pelas empresas para que o processo de cadastro seja o mais transparente possível, sendo possível em alguns minutos um sistema de segurança funcionando, porém com uma grande possibilidade da entrada de usuários maliciosos.

Para uma avaliação mais aprofundada foi utilizada a ferramenta *Nessus*, que possibilitou a análise e verificação de vulnerabilidades do sistema. Conforme a

Figura 18, ao realizar a análise foi possível verificar que a ferramenta possuia uma falha onde em um dos diretórios utilizados a senha *root* estava sendo disponilizada, porém de forma criptografada (*Hash* da senha). No diretório, o seguinte texto pode ser encontrado:

*root:absxcfbgXtb3o:0:0:root:/:/bin/sh*

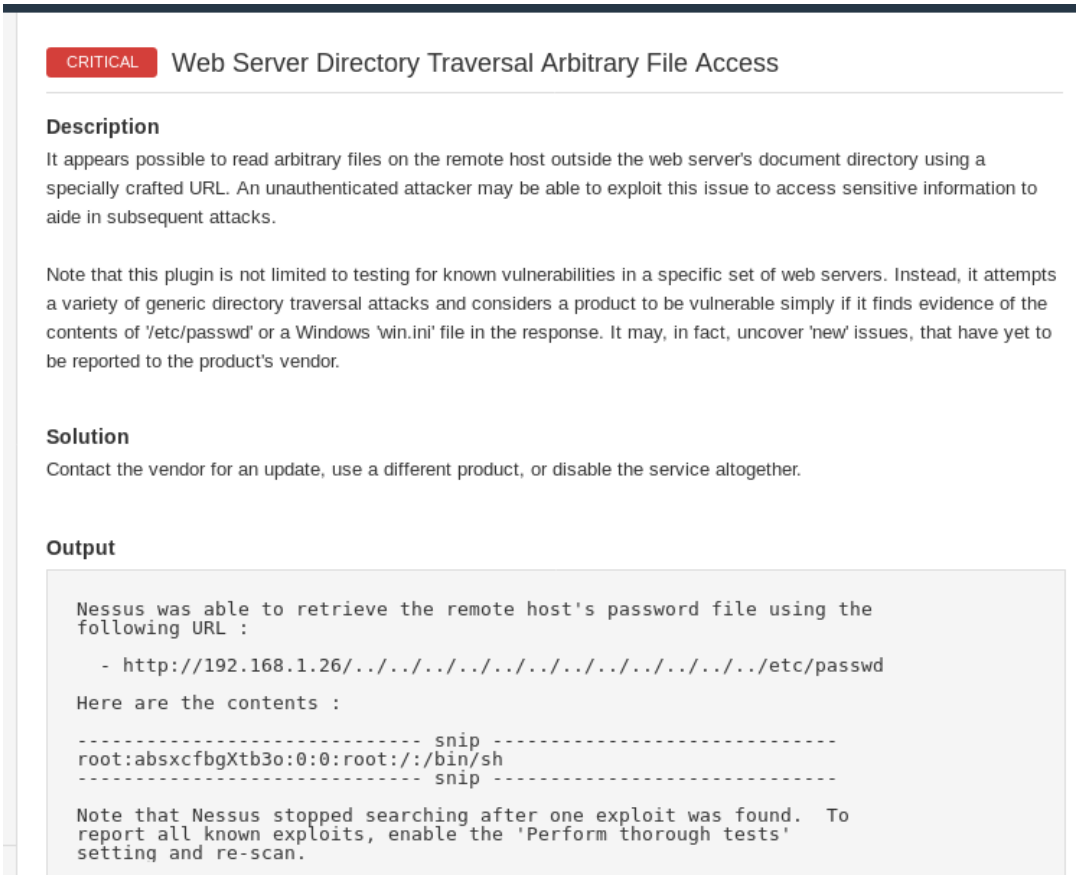


*Figura 18 - Verificação de vulnerabilidades com Nessus*

Para a quebra do *Hash,* a ferramenta *John the Ripper* foi utilizada. O sistema possibilita verificar através de força bruta se há a possibilidade da quebra da criptografia que está sendo utilizada. O processo de quebra de senha por força bruta realiza através de um algoritmo específico o teste de todas as possibilidades a fim de identificar o texto claro referente ao *Hash* que está sendo avaliado. Além disso, a ferramenta possui uma forma de quebra de senha a partir de um dicionário pré-configurado, dessa forma o proceso é realizado de forma a consumir menos tempo pois a ferramenta realizará o teste não mais com todas as combinações possíveis, mas com uma relação de senhas padrão de bases que podem ser encontradas em diversos sites que tratam do assunto.

Com o *John,* foi possível obter o resultado conforme evidenciado na Figura 19, verificando que o hash *absxcfbgXtb3o* corresponde a senha “xc3511”.

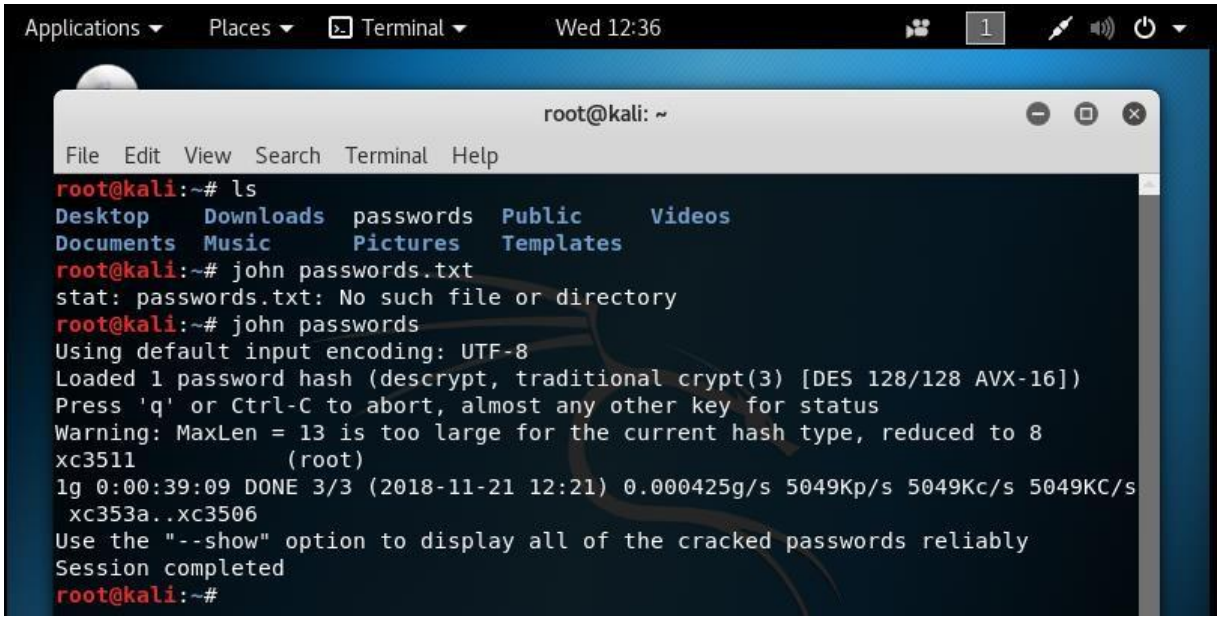


*Figura 19 - Quebra de Hash com o John the Ripper*

Para análise das portas que seriam interessantes de avaliação da senha encontrada, foi utilizada a ferramenta *NMAP* para verificação. Na Figura 20 pode-se verificar que a porta 23 (*Telnet)* está disponível para acesso.

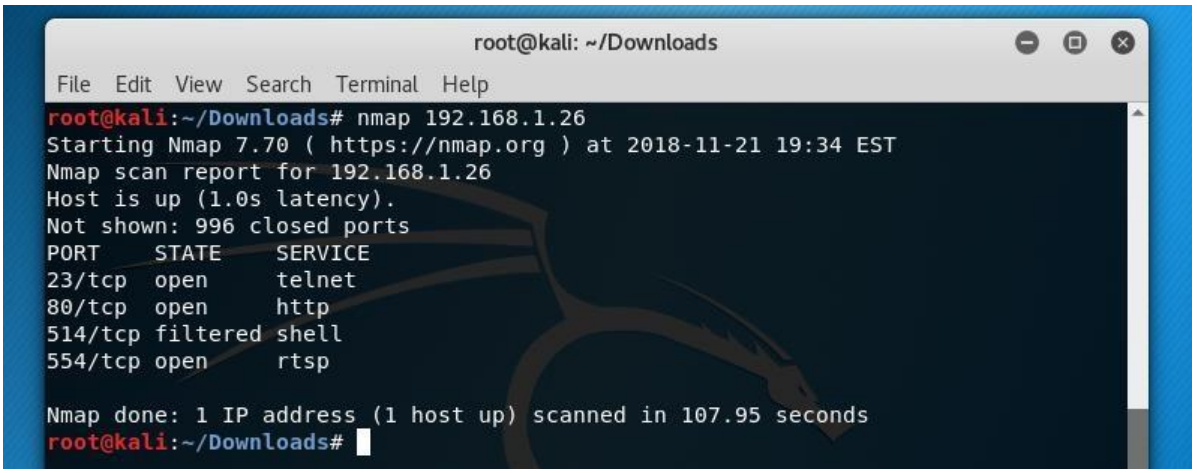


*Figura 20 - Aplicação*

O serviço *Telnet* permite a interface de terminais para a aplicação, com isso se houver o sucesso com a autenticação do usuário *root* seria viável se tornar *Owner* da ferramenta e de todos os atributos do sistema. A vulnerabilidade explorada permitiu que um atacante com acesso à rede pudesse obter acessos

administrativos (root) na ferramenta, conforme evidenciado na Figura 21:

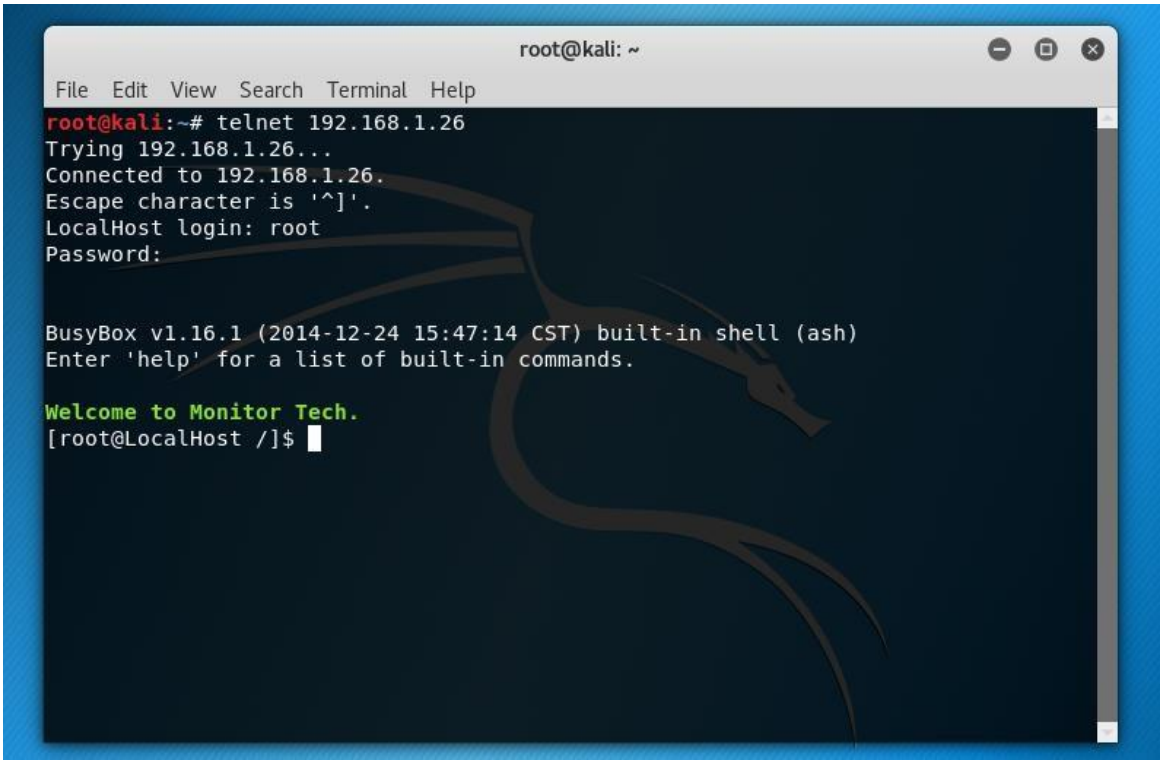


*Figura 21 - Conexão root no sistema*

Dessa forma, é possível observar o ambiente simulado da seguinte forma conforme Figura 22:

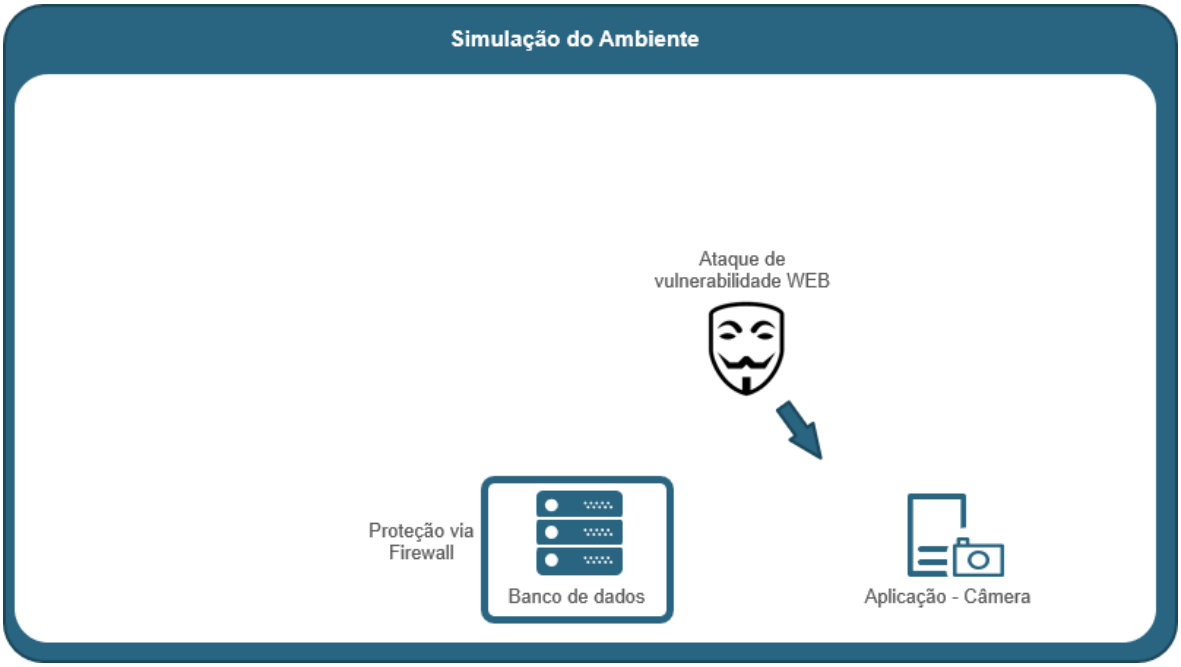


*Figura 22 - Estrutura de simulação 3*

### 4.3.2 Defesa

Avaliando o sistema integrado ao nosso ambiente corporativo, podemos validar que ao inseri-lo com uma disponibilidade de acesso via HTTP (porta 80), abrimos brecha para a posse e descoberta da senha do super usuário (Usuário *root).*

Para que haja uma possibilidade de defesa, é preciso utilizar o *Hardening* da ferramenta*.* O processo de *Hardening* de um determinado sistema é realizar o desligamento de recursos que não são utilizados ou possuem algum tipo de vulnerabilidade, para que o mesmo não possa ser explorado caso seja identificado.

No caso, o sistema não disponibiliza o *Hardening* total, porém é possível limitar o IP do serviço inserindo um cabo *Ethernet CrossOver* em um outro computador, ao invés de conectado diretamente ao roteador. Realizando o processo dessa forma, na Figura 23 é possível verificar que no servidor que o serviço está alocado, há visibilidade do serviço:

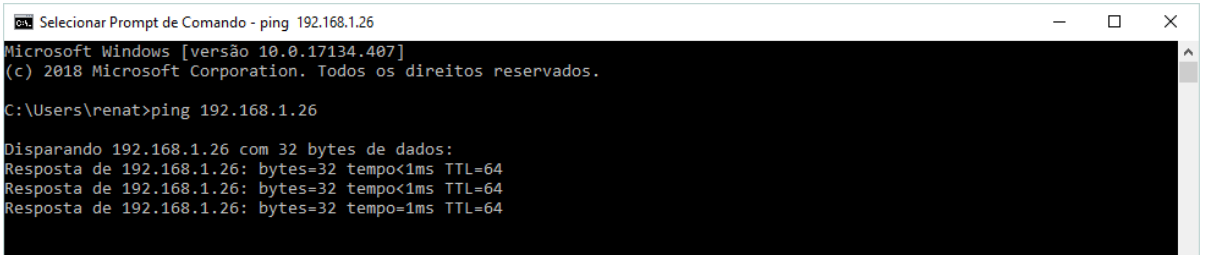


*Figura 23 - teste de conexão com o serviço com sucesso*

Com isso eliminamos as vulnerabilidades encontradas pela disponibilização do IP para acesso, porém é preciso disponibilizar um novo meio para acesso das imagens.

Para que a informação seja disponibilizada e de forma segura, foi necessária a criação de uma ferramenta para gestão da informação, autenticação e autorização dos usuários para acesso à informação.

Utilizamos o *Active DIrectory* para que o processo fosse realizado, sendo assim o usuário que desejasse a informação deveria ter uma identidade dentro do *AD,* bem como alocado em um grupo previamente definido.

Criamos então um servidor do *AD* para o processo de autorização e autenticação. O grupo "*G_CAM_ACCESS"* foi criado para o processo de autorização conforme a Figura 24. Da mesma forma, o usuário AZIDOI foi criado e o grupo foi liberado:

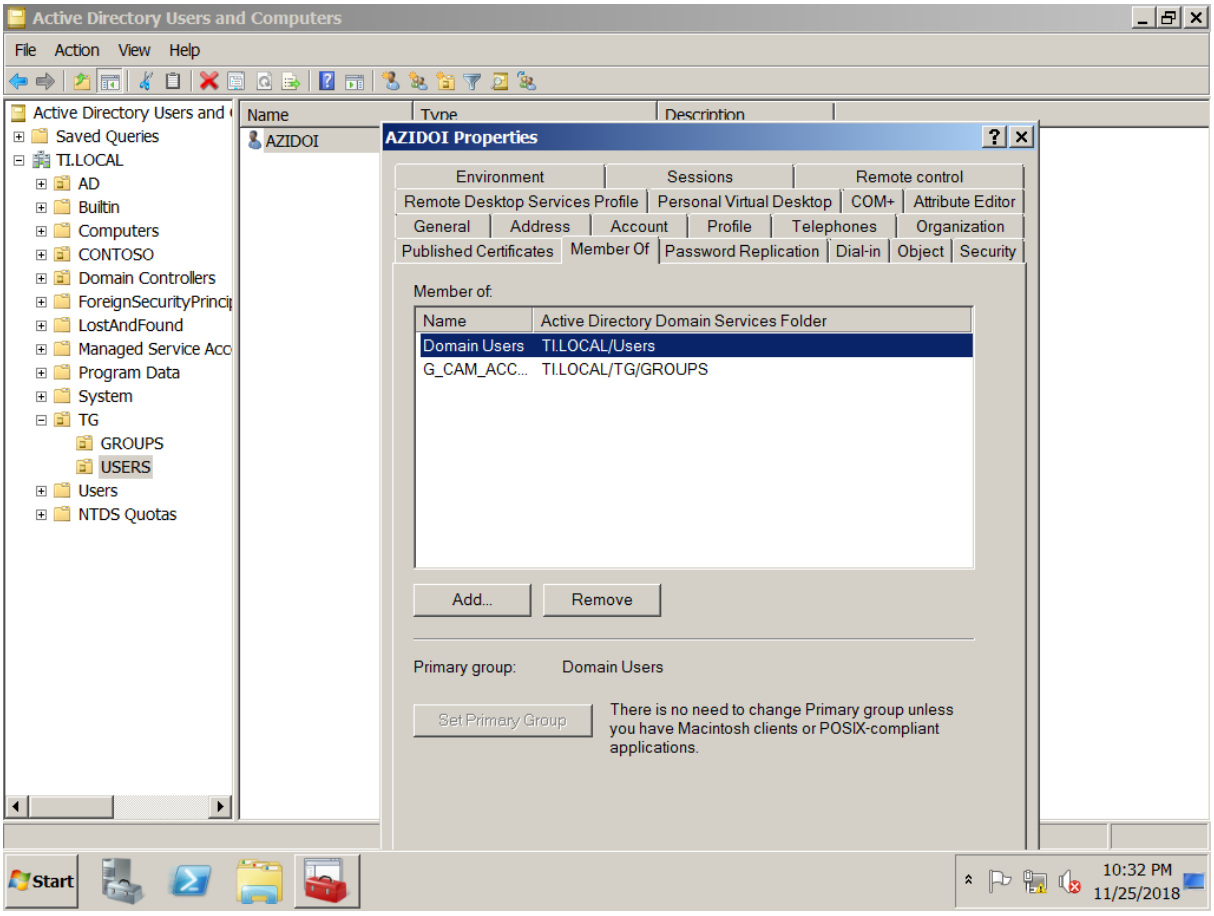


*Figura 24 - Liberação de acesso*

Dessa forma uma nova aplicação de autenticação e autorização foi criada, utilizando os grupos registrados via AD e usuários cadastrados na ferramenta (O código fonte da aplicação criada está no anexo "Código fonte Aplicação Gestão Câmeras"). Com uma comunicação com o Servidor, realizamos a autenticação do usuário a partir de seu usuário e senha:

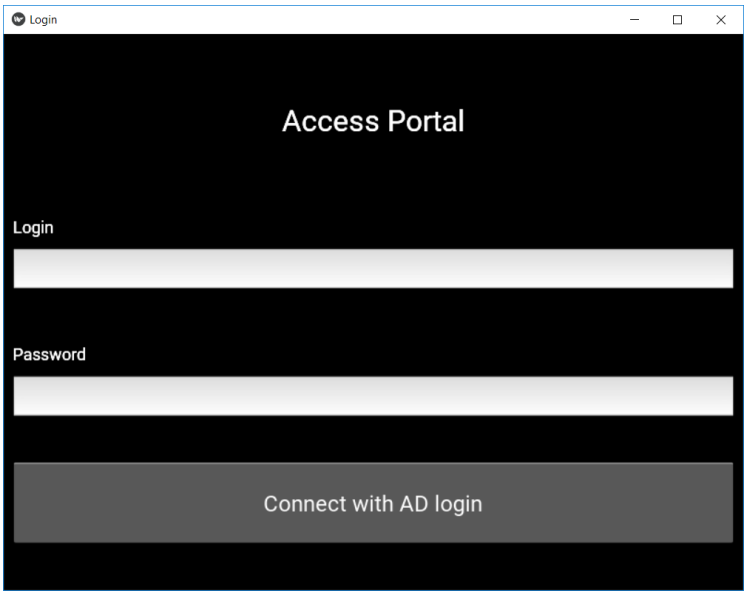


*Figura 25 - Plataforma de acesso*

Como teste, ao se realizar o acesso com usuário e senha incorretos, o seguinte erro ocorre:

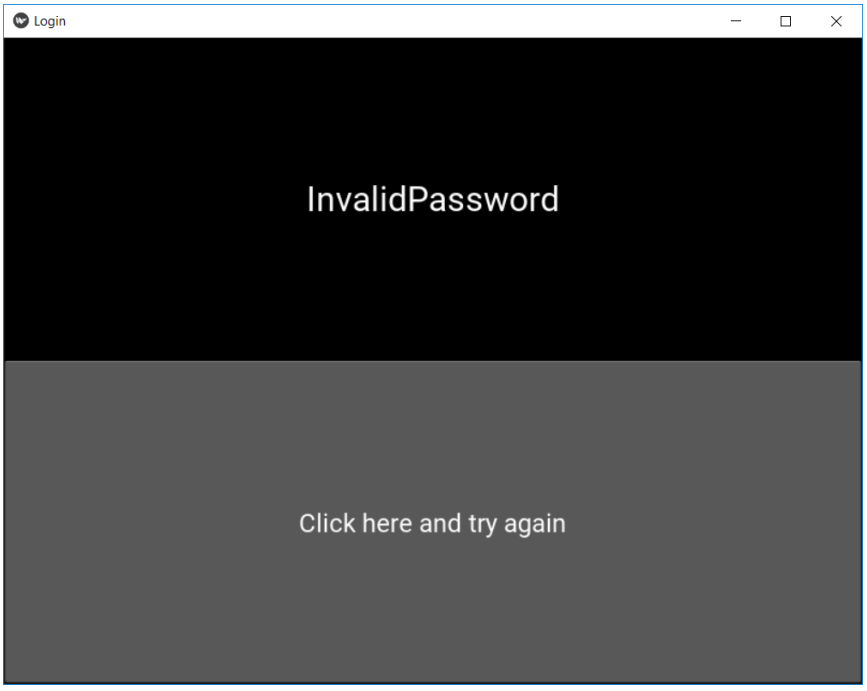


*Figura 26 - Credenciais invalidas*

Porém, quando o usuário se autentica corretamente, uma nova página é disponibilizada:

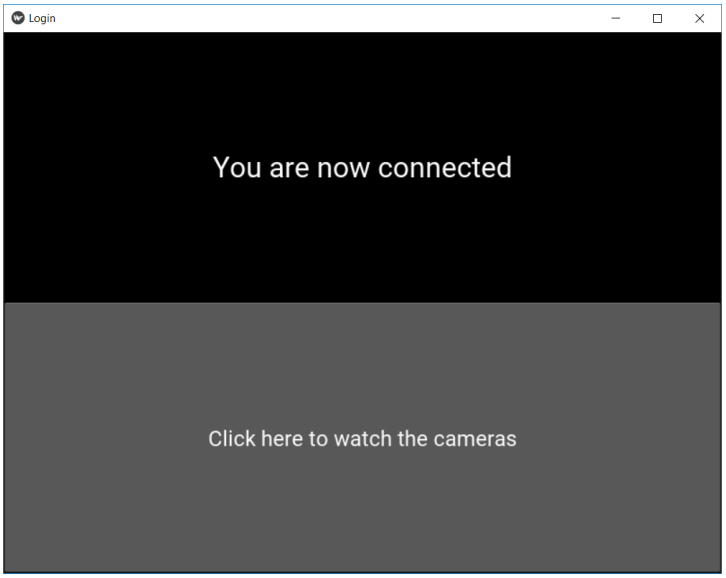


*Figura 27 - Sucesso na autenticação*

Foi criado também um usuário que não possui a permissão via *AD* para teste de acesso. Ao realizar a autenticação do novo usuário, a seguinte mensagem surge para o usuário:

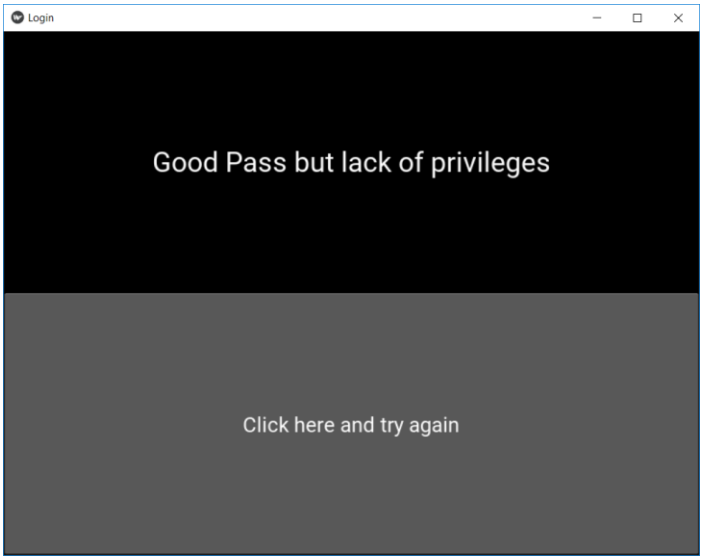


*Figura 28 - Falha na autorização de usuário*

Dessa forma, é possível observar o ambiente simulado da seguinte forma conforme Figura 29.A partir de um ataque analisando as vulnerabilidades de uma plataforma via *scan,* foi construindo um *Front-End* para gestão de autenticação e autorização de usuários, utilizando o *Active Directory* para esse processo.

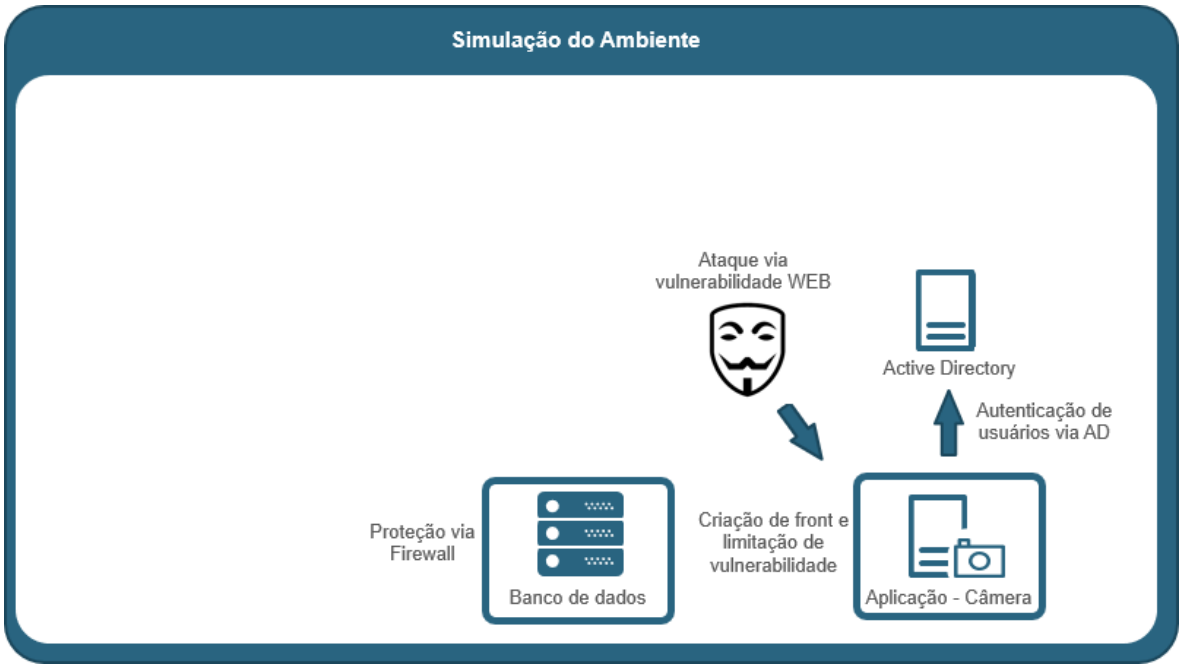


*Figura 29 - Estrutura de simulação 4*

## 4.4 Rede Local (WLAN)

### 4.4.1 Ataque

Para o ataque as redes com protocolo de segurança WEP, foi necessário utilizar o programa *airodump-ng* para monitorar o roteador desejado a fim de coletar as informações necessárias. Conforme a Figura 30, é possível verificar a rede *Rede_Ent* configurada com o protocolo de segurança WEP:

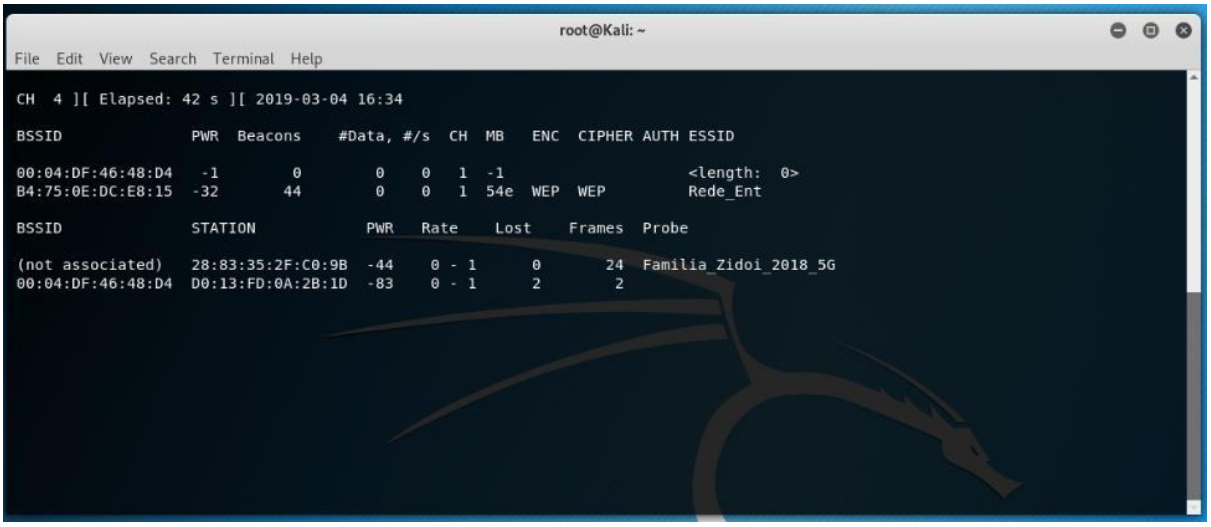


*Figura 30 – Analisando roteador WEP*

A partir disso, é possível utilizar a ferramenta *besside-ng* para realizar o ataque à rede na Figura 31.

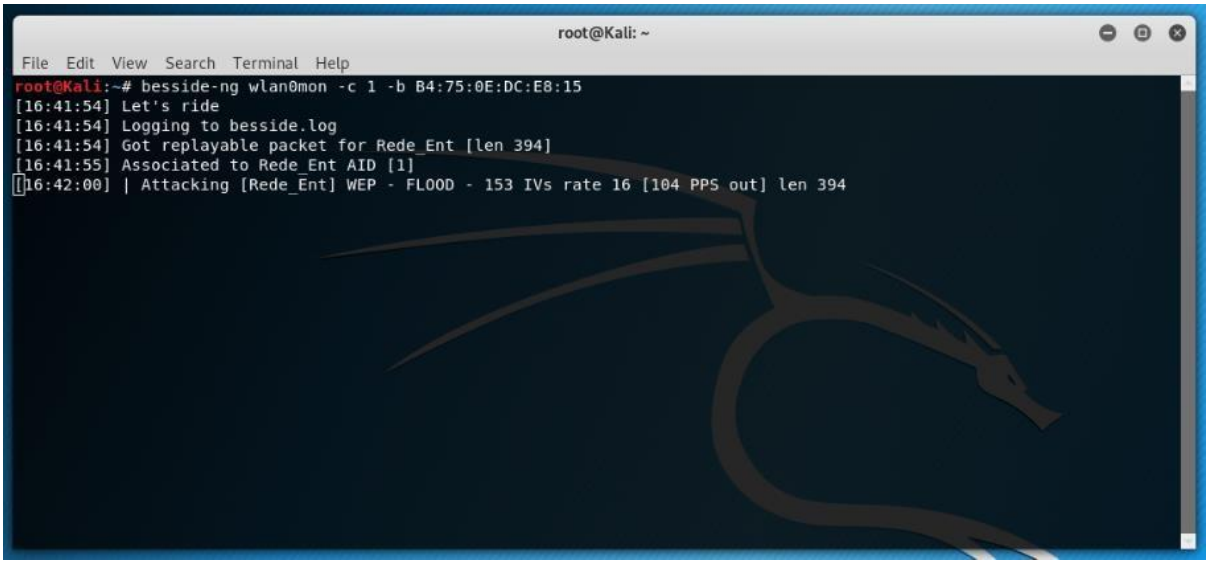


*Figura 31 - ataque com a ferramenta bessside-ng*

Dessa forma é possível realizar uma injeção de pacotes com finalidade de receber pacotes suficientes para as colisões IV. A partir de uma quantidade considerável de pacotes trocados, obtivemos a chave EA56CD6518 de acesso à rede conforme evidenciado na Figura 32.

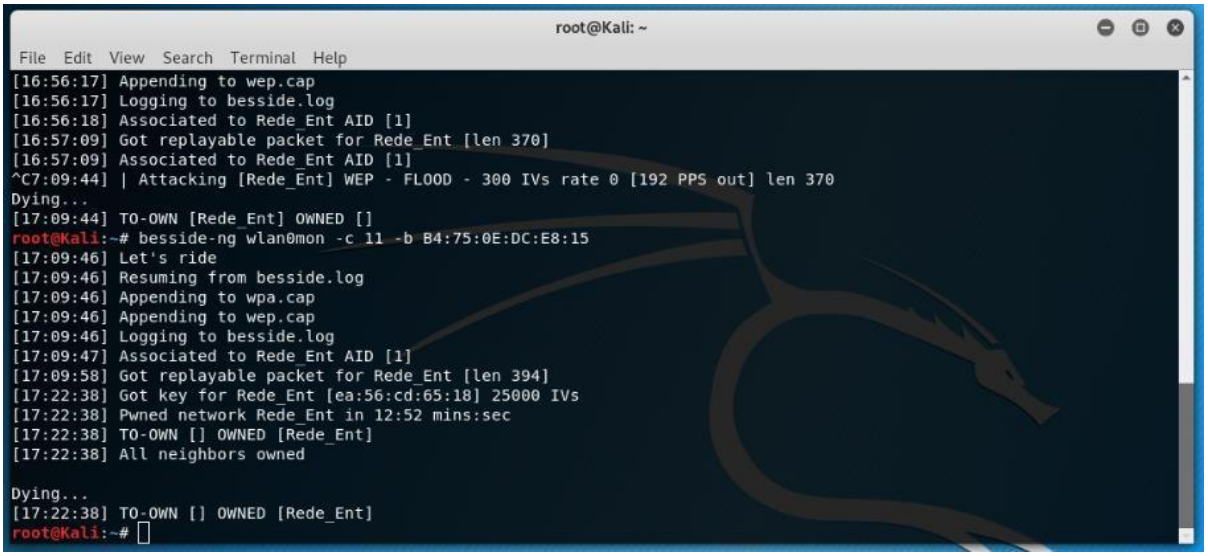


*Figura 32 – Senha de rede WLAN descoberta*

Dessa forma, é possível observar o ambiente simulado da seguinte forma conforme Figura 33. Uma exploração à vulnerabilidade do protocolo de rede foi simulada para acesso à rede local, com finalidade de obter uma visibilidade aos componentes que o ambiente possui e com isso permissão de comunicação com essas ferramentas.

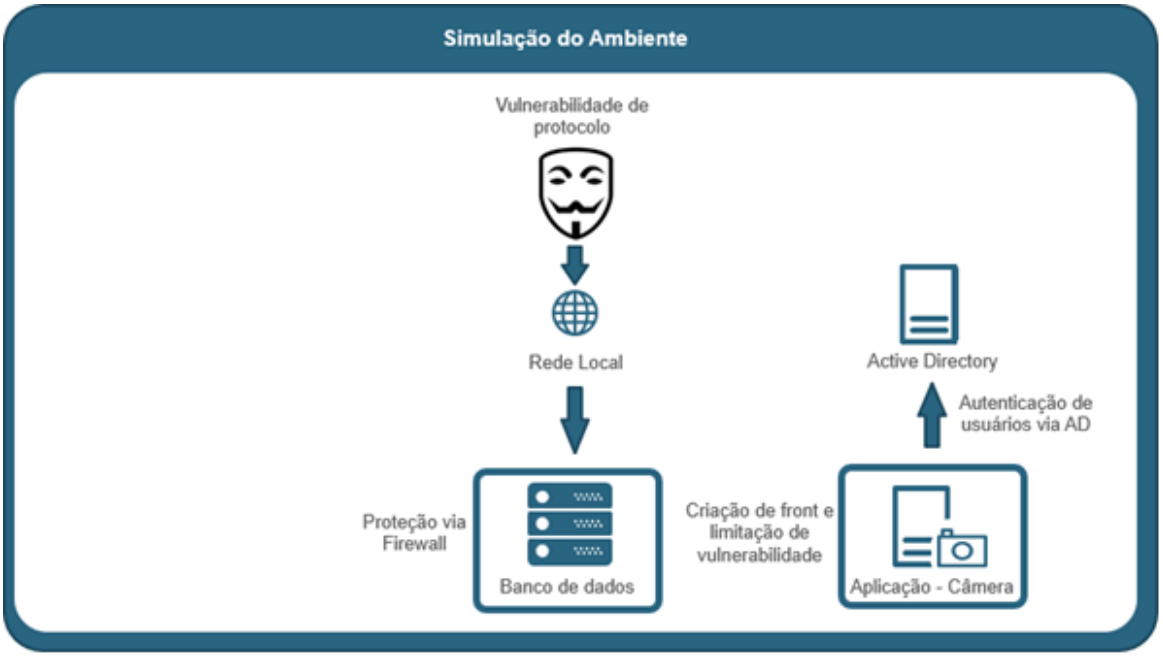


*Figura 33 - Estrutura de simulação 5*

### 4.4.2 Defesa

Conforme mencionado anteriormente, para haver um modelo seguro de autenticação e autorização à rede corporativa, é importante que apenas dispositivos previamente cadastrados no domínio da empresa tenham a possibilidade de acesso.

Sendo assim ataques que direcionam ao usuário, sua complexidade de senha, e *hash* de comunicação com o roteador são sanados pois a validação realizada independe do usuário que está utilizando a máquina em questão.

Para isso o roteador foi selecionado para o módulo WPA2 Enterprise, para que o IP do servidor Radius pudesse ser inserido conforme a Figura 34.

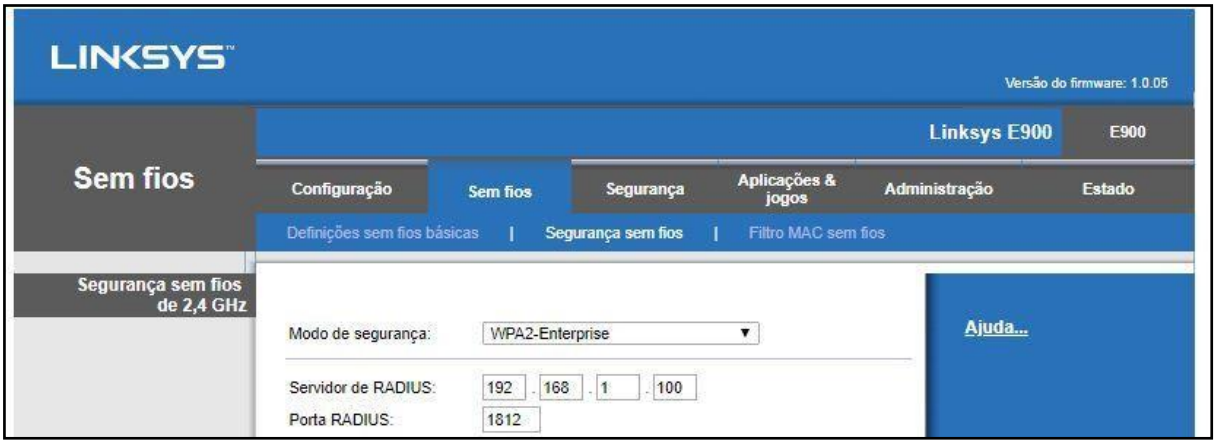


*Figura 34 - Configuração de Modo de Segurança Roteador*

No cenário em questão, o servidor utilizado possui o IP 192.168.1.100, instalado junto à aplicação do AD e a porta 1812 foi selecionada (Padrão de comunicação do protocolo RADIUS). Para realizar o cadastro das máquinas que deveriam ter o privilégio, o Active Directory foi necessário e um novo grupo “Wireless” foi criado conforme evidenciado na Figura 35.

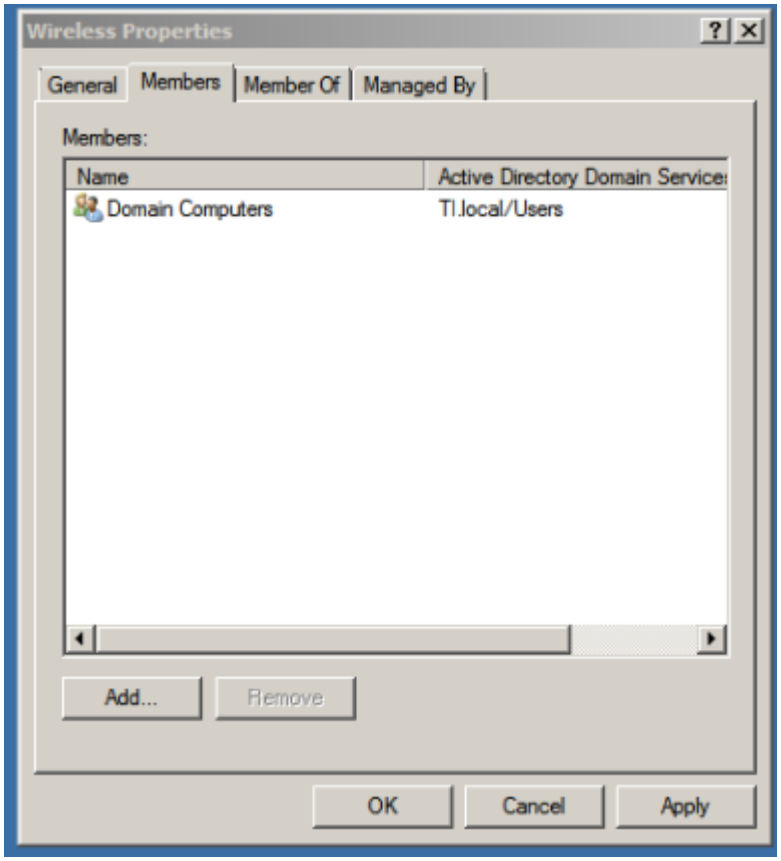


*Figura 35 - Grupo de permissão para Rede*

No grupo criado, foi vinculado o grupo “Domain Computers”, dessa forma todos os dispositivos que adentrassem ao domínio automaticamente teriam a permissão para acesso à rede. Após isso, conforme demonstrado na Figura 36 foi necessário realizar o cadastro do servidor Radius o vinculando ao SSID “Rede_Ent”.

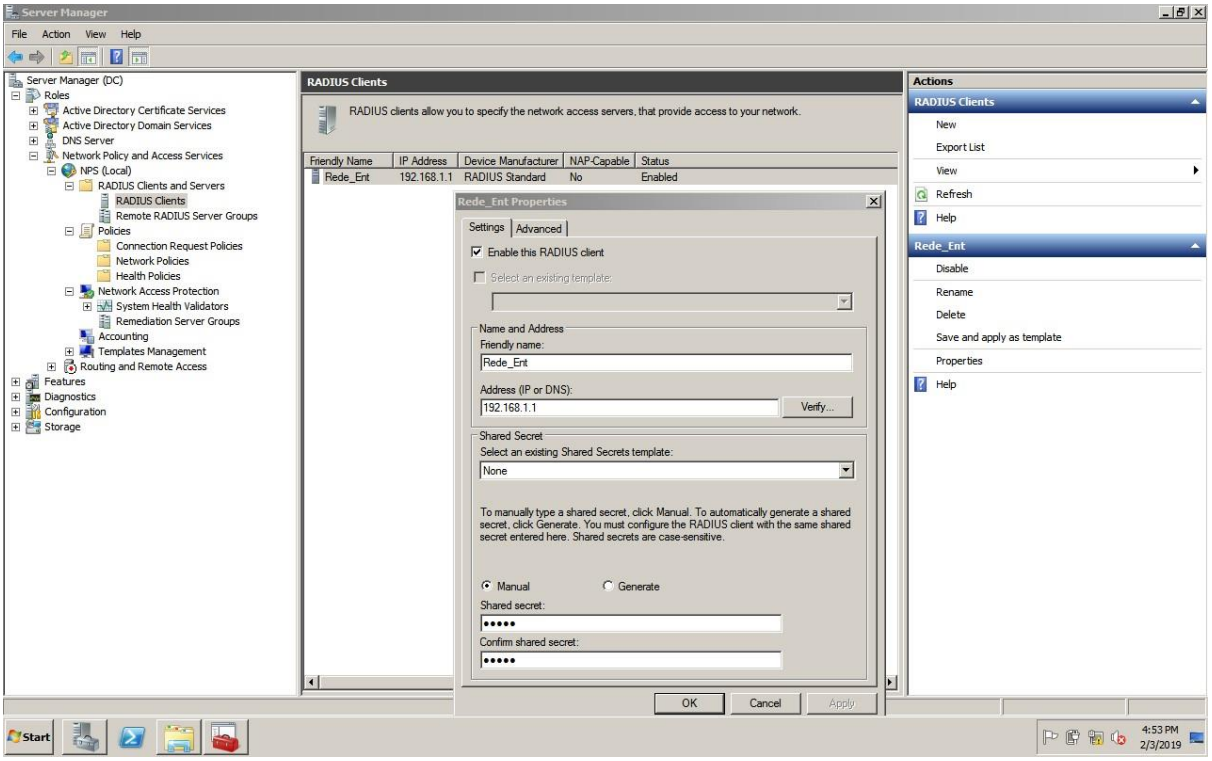


*Figura 36 - Configuração Radius Server*

Para finalizar as configurações da estrutura realizada, foi necessário que uma GPO (*Group Policy)* fosse criada para realizar a configuração que os dispositivos registrados estão com as certificações necessárias para a nova permissão. Conforma a Figura 37, o modo de autenticação criado para o uso da rede local foi a partir de computadores cadastrados previamente no domínio, dessa forma a rede *Rede_Ent* só teria a possibilidade de uso a partir desses dispositivos.

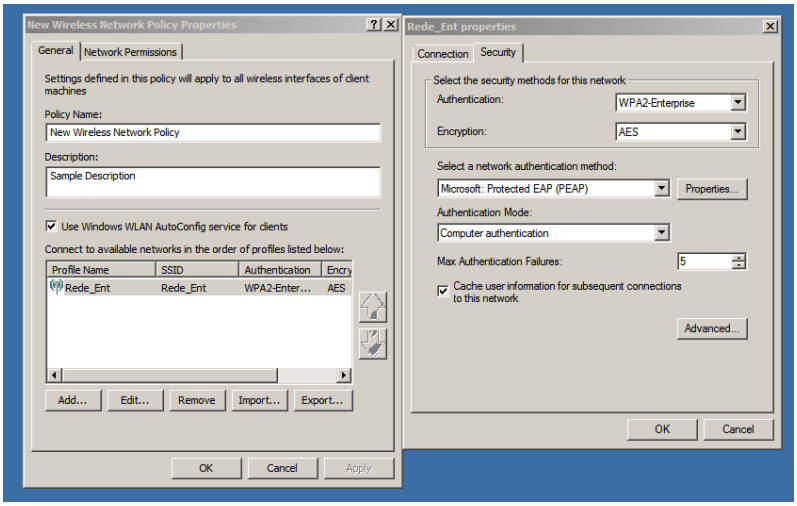


*Figura 37 - Configuração de GPO para os dispositivos*

Com todos os parâmetros configurados, na Figura 38 é verificado que um dispositivo cadastrado no domínio *TI.local* tem a permisssão de realizar a conexão na rede sem a necessidade de cadastro de uma senha para uso, devido seu registro de dispositivo estar permissionado a partir do grupo no *AD.* Quando essa permissão não é realizada, a solicitação retorna com um erro de “Não é possível conectar-se a esta rede”.

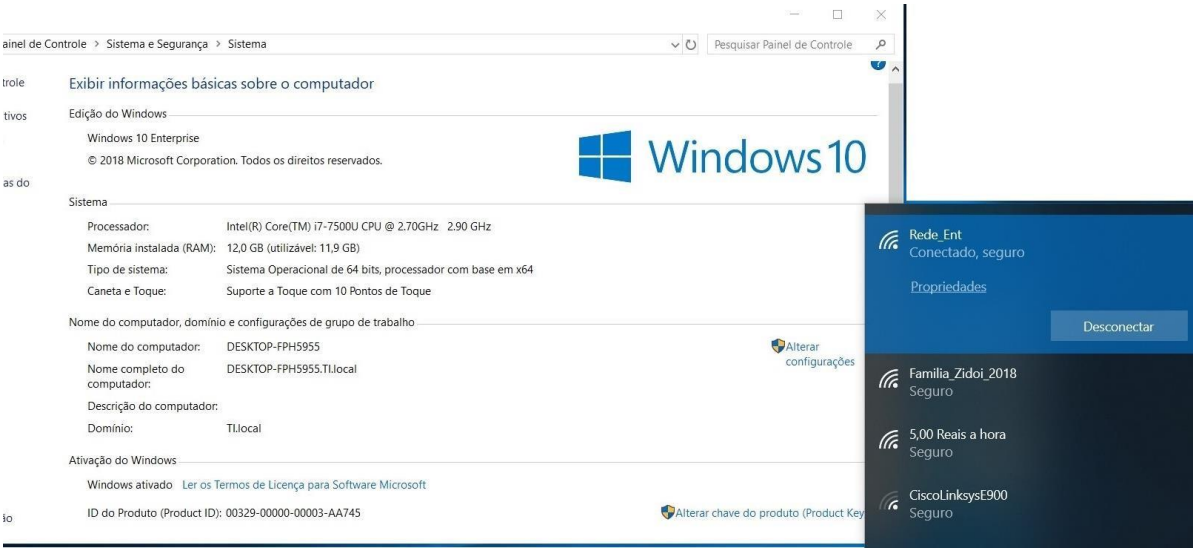


*Figura 38 - Conexão realizada para máquina registrada em domínio*

Dessa forma, é possível observar o ambiente simulado da seguinte forma conforme Figura 39. Para mitigação de exploração de vulnerabilidades em protocolo de redes, foi utilizado um modelo onde o dispositivo deveria estar autenticado e autorizado no *Active Directory.*

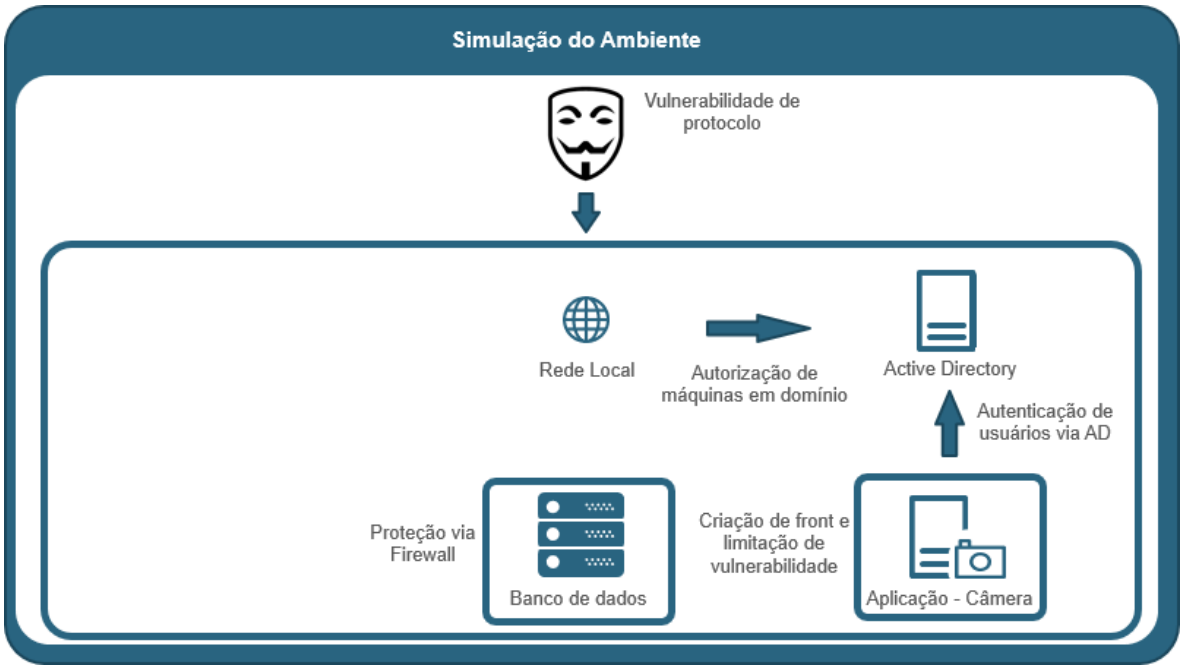


*Figura 39 - Estrutura de simulação 6*

## 5. Conclusão

Conforme descrito na evolução do projeto é possível verificar uma maior maturidade no ambiente corporativo no que tange aos recursos disponíveis para construção das defesas dos ataques mencionados. Verificamos um maior nível de disponibilidade da informação ao estabelecer regras para as conexões sendo realizadas a fim de evitar conexões maliciosas à ferramenta, um maior controle na autenticidade dos usuários e mitigação de vulnerabilidades inserindo um processo de autenticação e autorização de usuários para acesso à informação e uma avaliação de dispositivos conectando à rede estabelecendo um processo de eliminação de senha única para acesso à rede local.

Além disso outros inúmeros ataques podem ser realizados inserindo novas ferramentas ao ambiente, dessa forma é necessário que os conceitos de Segurança da Informação e suas implementações sejam avaliadas de forma tempestiva, novos modelos de controle sejam instaurados com finalidade de minimizar possíveis riscos de brechas ou vazamentos de dados.

O intuito principal do trabalho foi simular todos os ataques realizados com finalidade de obter um maior conhecimento em como os mesmos são realizados em sistemas reais, analisando sua efetividade e buscando conhecimento sobre invasões para um melhor entendimento de como mitigar essas falhas, e com essa geração de informação implementar as melhores práticas de mercado no ambiente simulado vulnerável.

Conforme já mencionado em capítulos anteriores, grande parte dos ataques efetivos realizados atualmente são de vulnerabilidades já conhecidas pela comunidade, sendo assim é necessário que haja um maior aprimoramento nos controles realizados pelas grandes empresas para que as vulnerabilidades já descobertas não sejam efetivas.

O trabalho de um analista de Segurança da Informação é entender de forma técnica as ações e processos realizados por atacantes mal intencionados, sendo assim é necessário que o mesmo esteja com seu nível de conhecimento sempre atualizado pois há sempre novas falhas em sistemas que são descobertas e precisam ser remediadas, e é função do analista de SI buscar todo tipo de conhecimento para estar à frente da defesa das corporações em que está alocado.

## 6. Referências

**[1]** Parmy Olson. **Nós Somos Anonymous. Por Dentro do Mundo dos Hackers.** Novo Século 1ª edição,2014.

[2] Mike Chapple and David Seidl. **Comptia CSA+ Study Guide**. John Wiley & Sons Inc 1ª edição, 2017.

**[3]** Yuri Diógenes. **Certificação Security+: Da prática para o exame SYO-401.** Novaterra 1ª edição, 2016.

**[4]** Georgia Weidman. **Testes de Invasão - Uma introdução prática ao hacking.** Novatec Editora 1ª edição, 2017.

[5] Fernando S. Meirelles. **Mercado Brasileiro de TI e Uso nas Empresas,** 2017 Disponível em:<eaesp.fgv.br/sites/eaesp.fgv.br/files/noticias2017gvcia.docx>

[6] Gustavo Zuccherato. **Indústria e Segurança da Informação: as principais ameaças de 2018,** 2018 Disponível em:<https://revistadigitalsecurity.com.br/artigo-industria-e-seguranca-da-informacao-as-principais-ameacas-de-2018/>

[7] PNAD IBGE. **cerca de 70% dos domicílios têm acesso à Internet,** 2018 Disponível em <http://www.abranet.org.br/Noticias/PNAD-IBGE:-cerca-de-70%25-dos-domicilios-tem-acesso-a-Internet-1787.html?UserActiveTemplate=site&UserActiveTemplate=mobile#.W3mBWOhKhPY>

[8] **O que você precisa saber sobre o PCI DDS 3.2,** 2016 Disponível em <http://blog.varonis.com.br/o-que-voce-precisa-saber-sobre-o-pci-dds-3-2/>

[9] **Nikto: Uma Ferramenta Open Source para Análise de Vulnerabilidades em Servidores Web,** 2015 Disponível em <http://www.lbd.dcc.ufmg.br/colecoes/freerbase/2015/004.pdf>

[10] SIRLEI LOURDES BACH. **Contribuição do Hacker para o desenvolvimento tecnológico da informática,** 2001 Disponível em <https://repositorio.ufsc.br/bitstream/handle/123456789/82176/184565.pdf?sequence=1>

[11] Todd Bey. **Understanding Multi-Facto authentication for PCI Compliance,** 2018 Disponível em <https://www.swc.com/blog/security/understanding-multi-factor-authentication-pci-compliance>

## Apêndice: Código fonte Aplicação Gestão Câmeras

```
from kivy.app import App
from kivy.properties import StringProperty
from kivy.uix.screenmanager import ScreenManager, Screen, SlideTransition
import os
from ldap3 import Server, Connection, ALL
import sys
import string
from    ldap3    import    Server,    Connection,    ALL,    NTLM,    ALL_ATTRIBUTES,
ALL_OPERATIONAL_ATTRIBUTES, AUTO_BIND_NO_TLS, SUBTREE
from ldap3.core.exceptions import LDAPCursorError

s = Server(host='192.168.36.132', port=389, use_ssl=False, get_info='ALL')
group = 'G_CAM_ACCESS'
format_string = '{:40} {}'

from connected import Connected
from invalidpass import Invalidpass
from passwtgroup import Passwtgroup

class Login(Screen):
   def do_login(self, loginText, passwordText):
       app = App.get_running_app()

       app.username = loginText
       app.password = passwordText
       print("TI"+"\\"+loginText)
       c    =    Connection(s,    user="TI"+"\\"+loginText,    password=passwordText,
auto_bind='NONE', version=3, authentication='SIMPLE',
                        client_strategy='SYNC',                    auto_referrals=True,
check_names=True, read_only=False, lazy=False,
                        raise_exceptions=False)
       if c.bind():

           c.search('dc={},dc=local'.format("TI"), '(objectCategory=person)',
                    attributes=[ALL_ATTRIBUTES, ALL_OPERATIONAL_ATTRIBUTES])

           for e in sorted(c.entries):
               try:
                   desc = e.memberOf
               except LDAPCursorError:
                   desc = ""

               if loginText in str(e.name):
                   if 'G_CAM_ACCESS' in str(desc):
                       self.manager.transition = SlideTransition(direction="left")
                       self.manager.current = 'connected'

                   else:
                       self.manager.transition = SlideTransition(direction="left")
                       self.manager.current = 'passwtgroup'
```

```
        else:
            self.manager.transition = SlideTransition(direction="left")
            self.manager.current = 'invalidpass'

        app.config.read(app.get_application_config())
        app.config.write()

    def resetForm(self):
        self.ids['login'].text = ""
        self.ids['password'].text = ""
class LoginApp(App):
    username = StringProperty(None)
    password = StringProperty(None)
    def build(self):
        manager = ScreenManager()
        manager.add_widget(Login(name='login'))
        manager.add_widget(Connected(name='connected'))
        manager.add_widget(Invalidpass(name='invalidpass'))
        manager.add_widget(Passwtgroup(name='passwtgroup'))
        return manager
    def get_application_config(self):
        if(not self.username):
            return super(LoginApp, self).get_application_config()

        conf_directory = self.user_data_dir + '/' + self.username

        if(not os.path.exists(conf_directory)):
            os.makedirs(conf_directory)

        return super(LoginApp, self).get_application_config(
            '%s/config.cfg' % (conf_directory)
        )

if __name__== '_main_':
    LoginApp().run()
```

Baseado no Código:

< https://gist.github.com/hanachan1026/37de8d567a325cca6c5064a2eac1f378>