CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

Recredenciado pela Portaria Ministerial nº 1.162, de 13/10/16, D.O.U. nº 198, de 14/10/2016
AELBRA EDUCAÇÃO SUPERIOR - GRADUAÇÃO E PÓS-GRADUAÇÃO S.A.

Cryslei Ferreira Gomes

PROPOSTA DE IMPLEMENTAÇÃO DE UMA CAMADA DE SEGURANÇA EM UM SERVIDOR DE REDE: SERVIÇOS PROFTPD, APACHE HTTP SERVER E OPENSSH

Palmas – TO

2020

Cryslei Ferreira Gomes

PROPOSTA DE IMPLEMENTAÇÃO DE UMA CAMADA DE SEGURANÇA EM UM SERVIDOR DE REDE: SERVIÇOS PROFTPD, APACHE HTTP SERVER E OPENSSH

Projeto de Pesquisa elaborado e apresentado como requisito parcial para aprovação na disciplina de Trabalho de Conclusão de Curso II (TCC II) do curso de bacharel em Ciência da Computação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Profa. M.e Madianita Bogo Marioti.

Palmas – TO

2020

Cryslei Ferreira Gomes

PROPOSTA DE IMPLEMENTAÇÃO DE UMA CAMADA DE SEGURANÇA EM UM SERVIDOR DE REDE: SERVIÇOS PROFTPD, APACHE HTTP SERVER E OPENSSH

Projeto de Pesquisa elaborado e apresentado como requisito parcial para aprovação na disciplina de Trabalho de Conclusão de Curso II (TCC II) do curso de bacharel em Ciência da Computação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Profa. M.e Madianita Bogo Marioti.

Aprovado em: _____/_____/_____

BANCA EXAMINADORA

_____

Profa. M.e Madianita Bogo Marioti

Orientador

Centro Universitário Luterano de Palmas – CEULP

_____

Prof. Esp. Fábio Castro Araújo

Centro Universitário Luterano de Palmas – CEULP

_____

Profa. Dra. Parcilene Fernandes de Brito

Centro Universitário Luterano de Palmas – CEULP

Palmas – TO

2020

AGRADECIMENTOS

Agradeço primeiramente a Deus por tudo, pela saúde, paciência, pois sem ele eu não teria conseguido concluir este trabalho, agradeço em especial a minha mãe Luciana Alves Ferreira e ao meu pai Pedro Gomes por todo apoio que deram para chegar até aqui, o orgulho que tenho de vocês é imensurável. Agradeço aos meus amigos e professores que sempre me apoiaram a continuar, e agradeço em especial a minha orientadora Madianita Bogo Marioti pela paciência que teve ao longo desses 1 ano e meio, desde a matéria de estágio até a conclusão da minha graduação, o aprendizado que tive no decorrer deste projeto, serão levados adiante. Obrigado a todos por tudo !

RESUMO

GOMES, Cryslei Ferreira. **“Proposta de implementação de uma camada de segurança em um servidor de rede: serviços ProFTPD, Apache HTTP Server e OpenSSH”**. 2020. Trabalho de Conclusão de Curso (Graduação) - Ciência da Computação, Centro Universitário Luterano de Palmas, Palmas/TO, 2020.

Este trabalho tem como objetivo apresentar uma proposta de implementação de uma camada de segurança em um servidor de rede que utiliza os serviços ProFTPD, OpenSSH e Apache HTTP *Server*. No estudo, foi configurado o ambiente de homologação que utiliza serviços inseguros ou possui fragilidades quando não configurados de forma adequada, fragilidades que, quando exploradas, podem comprometer um sistema, bem como as pessoas que os utilizam. Na finalidade de amenizar os riscos e impactos dos serviços descritos, uma solução viável apresentada neste trabalho é a implementação de mecanismos de segurança. Em uma máquina da rede, foi implementada a ferramenta de segurança Snort e realizados ataques sobre ela para validar a eficácia da ferramenta e aumentar o nível de segurança nos serviços presentes. E, por fim, foram descritos os testes realizados e o comportamento da rede, sendo proposta uma solução de um ambiente seguro que poderá ser replicado em redes que utilizam os serviços abordados.

Palavras-chave: Segurança, Snort, ProFTPD, OpenSSH, Apache HTTP *Server*

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

AC - Autoridade Certificadora

ASCII - American Standard Code for Information Interchange

CA - Certificados Assinados

CERT.BR -Centro de Estudos para Resposta e Tratamento de Incidentes em Computadores

CIA - Confidentiality, Integrity and Availability

FTP - File Transfer Protocol

HIDS - Host Intrusion Detection System

HTTP - Hypertext Transfer Protocol

HTTPS - Hypertext Transfer Protocol Secure

IANA - Internet Assigned Numbers Authority

IDS - Intrusion Detection System

IETF - Internet Engineering Task Force

IPS - Intrusion Prevent System

MDN - Mozilla Developer Network

NIDS - Network Intrusion Detection System

NIST - National Institute of Standards and Technology

NMAP - Network Mapper

RFC - Request For Comments

SSH - Secure Shell

SSL - Secure Sockets Layer

TLS - Transport Layer Security

WWW - World Wide Web

SUMÁRIO

1 INTRODUÇÃO

Diante aos avanços tecnológicos, informações pessoais e institucionais estão cada vez mais sendo disponibilizadas, armazenadas e trafegadas na rede. Essas informações, quando armazenadas e trafegadas em meios inseguros, podem ser exploradas por atacantes, a fim de obter informações sigilosas que podem comprometer uma empresa ou pessoa. Por isso, é importante buscar formas de manter essas informações protegidas de eventuais ataques.

Segundo Stallings (2014, p.10), ataque é "qualquer ação que comprometa a segurança da informação que pertence a uma organização" e pode ser classificado como ataque passivo e ataque ativo. O ataque passivo tem como objetivo monitorar e analisar as transmissões que ocorrem dentro da rede, sendo difícil de ser detectado, pois não há alteração nos dados e, normalmente, emissor e receptor não tomam conhecimento de um terceiro. O ataque ativo tem como objetivo modificar ou falsificar as transmissões que ocorram de dentro da rede, impedir este ataque é difícil, deve-se então realizar a detecção do ataque e, posteriormente, a recuperação dos danos causados.

É comum que as redes sejam alvos de atacantes, Tanenbaum e Wetherall (2012) afirmam que grande parte dos problemas de segurança são causados por pessoas maliciosas que estão tentando obter intencionalmente algum benefício, chamar atenção ou prejudicar alguém. O fato é que, quando esses ataques são realizados em uma rede vulnerável, a chance de terem êxito é alta.

Segundo GOMES (2020), uma rede demonstra-se vulnerável a ataques e apresenta fragilidades em seus serviços quando não se implementam políticas de segurança ou não utilizam ferramentas de segurança. Por exemplo, no trabalho desenvolvido no estágio por GOMES (2020), foi criado um servidor de rede que oferece os serviços ProFTPD, OpenSSH e Apache HTTP *Server* sem configuração de segurança e foram realizados ataques para a demonstração de vulnerabilidades nesses serviços. Foram utilizadas as ferramentas Nmap, Wireshark e Metasploit, para a realização de ataques de força bruta e captura de dados na rede, demonstrando as vulnerabilidades dos serviços configurados.

Com isso, é importante proteger essas informações trafegadas na rede de eventuais ataques que comprometam a confidencialidade e integridade dessas informações, bem como manter a disponibilidade da mesma, para que possam ser utilizadas sempre que usuários autorizados a desejarem. Desse modo, para garantir a confidencialidade, disponibilidade e integridade dos dados, é necessário implementar medidas que garantam a segurança dessas informações.

Redes que não possuem medidas de segurança implantadas ou em que as medidas de segurança não são configuradas adequadamente se tornam vulneráveis a ataques. Segundo Tanenbaum (2003), a maior parte dos problemas relacionados à segurança está relacionado com pessoas que tentam obter algum benefício, demonstrar suas habilidades de hacking ou prejudicar alguém. Isso prova que uma rede, seja ela corporativa ou doméstica, para estar segura não basta apenas que esteja livre de erros de programação.

É comum que em servidores sejam utilizados serviços de rede que oferecem vários recursos, que podem ser tanto de uso interno quanto externo. Servidores que implementam serviços como ProFTPD, OpenSSH e Apache HTTP Server são muito conhecidos e utilizados, pois oferecem recursos utilizados com frequência por pessoas e sistemas, como transferência de arquivos, disponibilização de web sites e acesso remoto, respectivamente. Por serem serviços bastante utilizados e, consequentemente, visados pelos atacantes, é importante garantir a segurança dos dados armazenados e trafegados contra eventuais ataques que possam ocorrer nesses serviços por pessoas mal intencionadas, aplicando conceitos e mecanismos que aumentem o grau de segurança na rede.

Nesse contexto, deu-se continuidade ao trabalho desenvolvido no estágio de Gomes (2020), foi desenvolvido um trabalho com objetivo de apresentar uma proposta e implementação de uma camada de segurança em um servidor de rede, com os serviços ProFTPD, OpenSSH e Apache HTTP *Server*, e demonstrar a eficiência das ferramentas de segurança indicadas. Espera-se que este trabalho sirva de referência para estudantes da área ou para profissionais que precisem implementar uma rede similar à apresentada neste trabalho.

2 REFERENCIAL TEÓRICO

Nesta sessão serão abordados os conceitos sobre segurança da informação, portas, protocolos HTTP, SSH, FTP, TLS e mecanismos de segurança IPS e IDS.

2.1 SEGURANÇA DA INFORMAÇÃO

Diante aos avanços tecnológicos, grande parte da população está conectada ao meio digital, gerando milhares de informações a cada instante. Informações que, quando trafegadas por meios inseguros em uma rede de computadores, como a *Internet*, pode levar a um grande problema no que se refere à segurança, como proteger essas informações?

Para garantir segurança para as informações trafegadas nas redes é necessário atender alguns princípios básicos de segurança. Segundo a NIST (*National Institute of Standards and Technology*), há três pilares de segurança que devem ser garantidos, a confidencialidade, a integridade e a disponibilidade, normalmente, chamados de **Tríade CIA** (do acrônimo em inglês para *confidentiality*, *integrity and availability*). Documentada pelos padrões da NIST, os três conceitos envolvem os objetivos fundamentais da segurança da informação definidos em cada categoria:

- Confidencialidade: proteção da privacidade de indivíduos e informações sigilosas;
- Integridade: garantia de que os dados não tenham sidos alterados, inseridos ou destruição imprópria de informação de forma não autorizada;
- Disponibilidade: garantia de que os sistemas operem a todo momento, de forma que os serviços estejam disponíveis para usuários autorizados.

A tríade representa os principais atributos de um sistema seguro, sendo que, para garanti-los deve-se implementar mecanismos de segurança, com controle de acesso e autenticação. Segundo a recomendação X.800 da *International Telecommunication Union* (1991), Controle de Acesso e Autenticação quando são implementados elevam o grau de segurança na rede, conceitos que são descritos como:

- Controle de Acesso: assegurar que somente pessoas autorizadas possam acessar o sistema, bem como controlar e limitar o acesso a usuários em um sistema;
- Autenticidade: garantir que as partes envolvidas na comunicação sejam autênticas, isso significa verificar se os usuários são o que dizem ser.

Para aumentar a segurança da rede, de forma a respeitar os pilares de segurança e oferecer controle de acesso e autenticação, é recomendado que sejam implementados

mecanismos de segurança na rede. Esses mecanismos permitem que apenas pessoas autenticadas consigam acessar a rede e quais dados podem ser acessados, de acordo com o tipo de usuário definido.

Neste trabalho serão aplicadas ferramentas de segurança para proteger os serviços ProFTPD, OpenSSH e Apache HTTP *Server*. Os protocolos que definem as características desses serviços serão abordados nas seções seguintes.

2.2 PORTAS

A “porta” atua como uma ponte de conexão entre máquinas em uma rede. Documentada na RFC 6335, existe um total de 65535 portas sendo distribuídas em três intervalos:

- *System Ports*: também conhecidas como *Known Port*s, sendo portas numeradas entre 0 a 1023 utilizadas para serviços mais rígidas que as demais e só serão concedidos sob os procedimentos de "Revisão IETF" ou " Aprovação IESG " descritos na RFC 5226.
- *User Ports*: portas entre 1024 a 49151, estão disponíveis para atribuições por meio da IANA e comumente utilizados para dispositivos externos;
- *Dynamic and/or Private Ports*: portas no intervalo de 49152 a 65535 foram reservadas especificamente para uso local e dinâmico. De acordo com a IANA, as portas entre essas faixas não devem ser usadas como um identificador de serviço.

Neste trabalho são mencionadas as portas utilizadas por padrão nos serviços ProFTPD, OpenSSH e Apache HTTP *Server*. Os protocolos que definem as características desses serviços serão abordados nas seções seguintes.

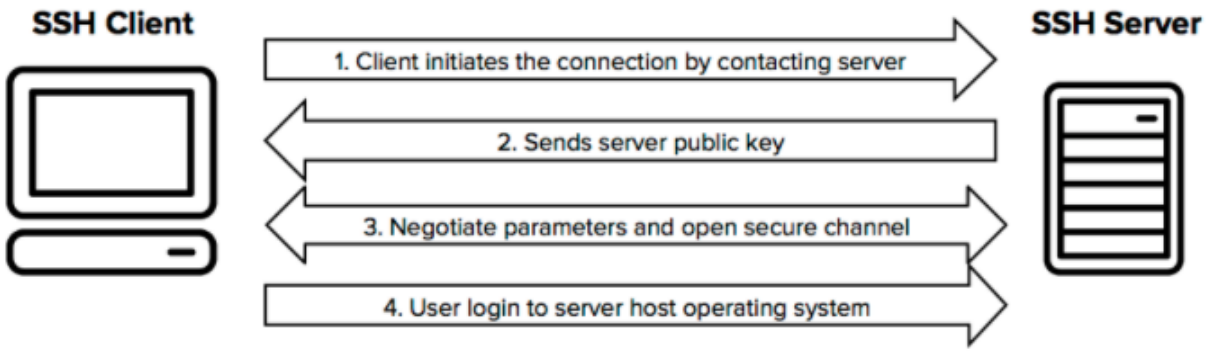
2.3 PROTOCOLO SSH

O SSH é um protocolo para acesso remoto, uma tecnologia que permite a interação entre duas máquinas que estejam conectadas à rede. O protocolo oferece a troca segura de dados, para isso, utiliza-se um canal seguro entre dois dispositivos de redes, que por padrão utilizam a porta 22. Os serviços que implementam esse protocolo são bastante utilizados em sistemas baseados no Unix para permitir acesso ao *Shell* dos servidores.

De acordo com Stallings (2014), Secure Shell (SSH) é um protocolo para as comunicações seguras, documentado na RFC 4251. Para isso, utiliza a criptografia dos dados, garantindo a confidencialidade e integridade dos dados na comunicação, mesmo em uma rede insegura como a internet. A primeira versão, SSH1, foi criada para substituir o TELNET, um

serviço de acesso remoto muito usado antes do SSH, e outros protocolos inseguros de acesso remoto.

Figura 1 - Funcionamento do protocolo SSH



fonte:(https://websitebuilding.biz/development/how-to-enable-ssh-on-a-raspberry-pi/)

A figura 1 apresenta o processo de comunicação entre o cliente e servidor através do protocolo SSH, no qual o cliente inicia o contato com o servidor (1), na sequência é realizado o processo de criptografia na comunicação dos dados (2 e 3) e, por fim, é efetuada a autenticação do cliente ao servidor (4).

Atualmente é mais utilizada a segunda versão SSH2, documentada como um padrão proposto nas RFCs 4250 a 4256 do IETF. Essa versão busca resolver uma série de falhas existentes no SSH1, entre elas, na versão 1 era possível inserir de forma não autorizada dados no meio do fluxo de dados criptografados, o que pode causar um alto risco à segurança dos dados, além de apresentar vulnerabilidade na autenticação. A nova versão do SSH2 é incompatível com a versão anterior, por usar um conjunto diferente de algoritmos aprimorados e mais fortes para criptografia e autenticação.

Stallings (2014) afirma que aplicações que implementam o SSH podem ser encontradas facilmente para a maioria dos sistemas operacionais, tornando-se um dos serviços mais utilizados para acesso remoto, e está se tornando um dos serviços mais difundidos para a tecnologia de encriptação fora dos sistemas embutidos.

2.4 PROTOCOLO FTP

O FTP é um protocolo utilizado em larga escala por profissionais da área e serviços para transferência de arquivos entre cliente e servidor. A comunicação é basicamente composta por uma fase de conexão, em que ocorre a autorização do usuário; a fase operação,

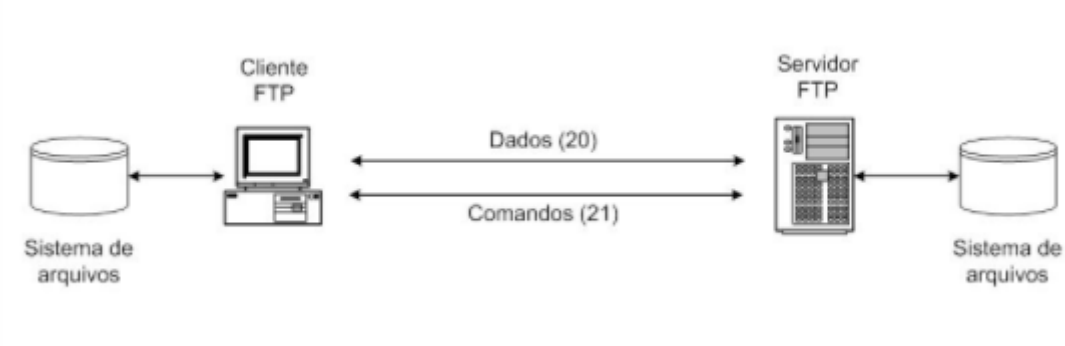
na qual são passados comandos ftp do cliente para o servidor; a fase da transferência de arquivos; e o encerramento da conexão, sendo responsabilidade do usuário fechar a conexão.

Segundo Chung (2014), é o protocolo mais usado para transferir arquivos para Internet, sendo uma alternativa para essa funcionalidade do protocolo HTTP. É documentado pela RFC 959 do IETF, sendo que, pelas definições do protocolo, o processo de autenticação entre cliente e servidor é realizado de forma insegura, já que não é definido mecanismo de criptografia na comunicação, cabendo ao administrador da rede configurar as opções de segurança, normalmente oferecidas pelos serviços que implementam o protocolo. Além de não prever segurança no processo de autenticação, por padrão, as mensagens são trocadas em texto plano desde o processo de conexão.

O protocolo usa duas conexões durante uma sessão: uma para controle e o outra para transferência de dados, sendo utilizadas as portas 20 e 21, que, conforme definida pela IANA (*Internet Assigned Numbers Authority*), como mostra a figura 2:

- Porta 20: conhecida como "*Data channel*", utilizada para a transferência dos dados, controle do fluxo e integridade dos dados;
- Porta 21: conhecida como "*Control Channel*", é utilizada para estabelecer e manter a comunicação entre o cliente e o servidor.

Figura 2 - Funcionamento do protocolo FTP



fonte: (https://slideplayer.com.br/slide/3221520/)

A Figura 2 apresenta o funcionamento do FTP, a comunicação se inicia a partir do momento em que o cliente solicita a conexão para o servidor FTP e o processo de conexão é realizado, utilizando, por padrão, a porta 21. Segundo KUROSE (2006), a porta 21 é usada para trocar informações de controle como *login* e *password*, mudar de diretório, solicitar ou enviar arquivos, enviando mensagens FTP no formato ASCII e utilizando comandos FTP. Já a transferência de arquivos é realizada pela conexão de dados, utilizando a porta 20 como

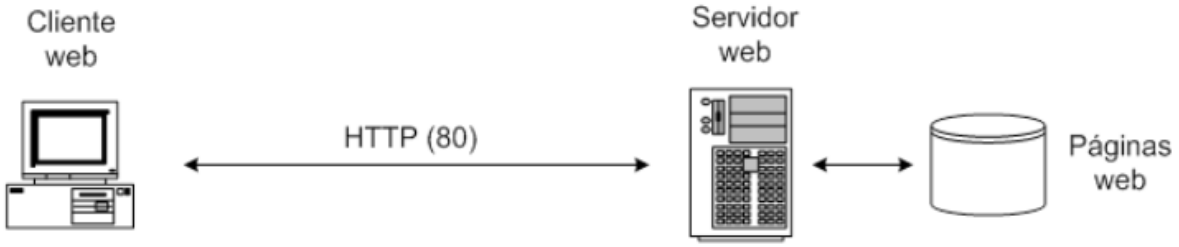
padrão. Por fim, a conexão entre cliente e servidor é encerrada, normalmente, através de um comando enviado utilizando a conexão de controle, porta 21.

2.5 PROTOCOLO HTTP

Documentado na RFC 7230, o HTTP é um protocolo de transferência de hipertexto, que pode ser usado para outras tarefas além da transferência de hipertexto, como buscar imagens, vídeos e publicar conteúdo em servidores. O HTTP é o protocolo usado e criado para a arquitetura da *World Wide Web* desde 1990, conforme a RFC 2068.

Kurose (2014) afirma que o protocolo HTTP é implementado em dois programas, denominado como cliente e o outro servidor, esses programas permitem a comunicação entre sistemas diferentes por meio da troca de mensagem HTTP. Documentada pela RFC 7230, os termos cliente e servidor fazem referência às funções específicas para cada conexão, por exemplo, um *browser* pode atuar como cliente que irá fazer as requisições e o servidor envia respostas ao *browser*. A figura 3 apresenta a comunicação entre o cliente e o servidor por meio do protocolo HTTP.

Figura 3 - Comunicação HTTP



fonte: (https://slideplayer.com.br/slide/3221520/)

A comunicação ocorre quando o cliente envia uma solicitação ao servidor HTTP, o servidor responde à solicitação de um cliente enviando uma ou mais mensagens HTTP de resposta, por padrão, utiliza-se a porta 80. Nas definições padrão do protocolo as mensagens trafegam em texto plano, de forma que é possível ler o conteúdo caso seja interceptado, tornando-se inseguro por não definir ações de segurança para os dados trafegados.

Além das informações dos arquivos de hipertexto que são transferidos entre cliente e servidor, as mensagens HTTP carregam várias informações em seu cabeçalho, como o

Content-Type e o *Cookie*. Descrito no site da MDN (*Mozilla Developer Network)*, o *Content-Type* especifica o tipo do conteúdo da resposta e o *Cookie* é usado para informar se duas solicitações vieram do mesmo navegador, mantendo um usuário logado, por exemplo. Essas informações podem ser úteis para os atacantes, quando exploradas.

Para que o servidor retorne as informações para o cliente, o cliente deve enviar informações para o servidor que, muitas vezes, são importantes e que deveriam ser sigilosas, por exemplo, quando os clientes preenchem formulários como o de *login* e *password* para conectar-se a um determinado serviço e que são enviadas ao um servidor via HTTP.

Assim, a característica do HTTP de não oferecer segurança por padrão é perigosa, por isso, mecanismos de criptografia podem ser combinados ao protocolo HTTP, como o protocolo TLS, que é descrito na seção seguinte.

2.6 PROTOCOLO TLS/SSL

O protocolo TLS fornece segurança de comunicação na internet e permite a comunicação entre cliente e servidor de uma forma projetada para evitar espionagem, adulteração ou falsificação de mensagens. O protocolo TLS veio para corrigir uma série de falhas do protocolo SSL v.3.0, documentado na RFC 7568, pois este apresentava fragilidades como na troca de chaves e no algoritmo utilizado.

De acordo com Barnes (2015) a versão SSL v.3.0 surgiu em 1996 mas foi descontinuada em Julho de 2015, dando lugar ao TLS 1.0 em 1999. O protocolo TLS não suporta o seu antecessor SSL, por definir uma série de modificações, entre elas, a combinação de algoritmos criptográficos, conforme documentado na RFC 7568.

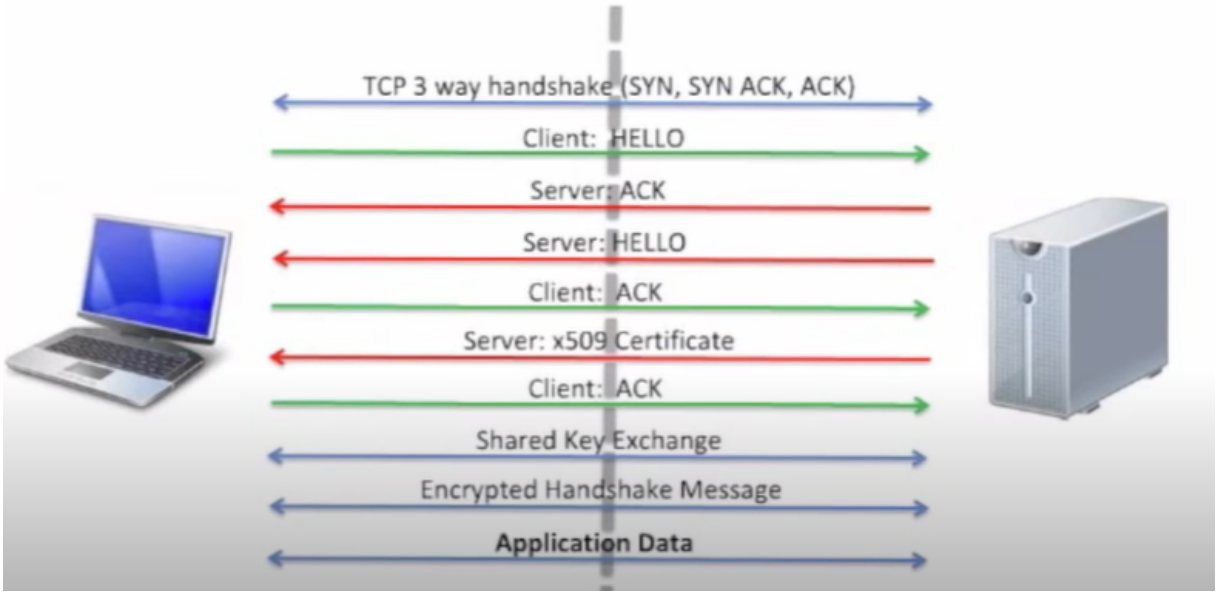
O objetivo principal do TLS é garantir uma comunicação segura entre dois pares, transportando os dados em um canal seguro. Conforme documentado na RFC 8446, o canal deve fornecer as seguintes propriedades:

- Autenticação: o lado do servidor do canal é autenticado, já a autenticação do lado do cliente é opcional, ou seja, o cliente utiliza as informações enviadas pelo servidor para autenticar o servidor, se o cliente está solicitando um recurso de servidor que requer autenticação do cliente, o servidor solicita o certificado do cliente. A autenticação pode acontecer por meio de criptografia assimétrica, algoritmo de assinatura digital ou a partir de uma chave pré-compartilhada simétrica;
- Confidencialidade: os dados trafegados pelo canal são visíveis apenas para o destinatário, pois o TLS utiliza o procedimento chamado de *handshaking*, onde o

cliente e servidor concordam em vários parâmetros utilizados para estabelecer a conexão segura;

- Integridade: os dados não podem ser modificados durante o envio por atacantes, pois fornece integridade de dados calculando um trecho da mensagem.

Figura 4 - Processo de comunicação SSL entre cliente e servidor



fonte:(https://www.youtube.com/watch?v=bGZ9HXxegno)

A figura 4 apresenta o processo de comunicação realizado pelo uso tls/ssl, em que: inicialmente, acontece a conexão entre cliente e servidor com o *three way handshaking;* na sequência o cliente envia um pacote *client HELLO* para o servidor, que contém todas as informações de criptografia que são suportados pelo cliente; o servidor confirma o recebimento do pacote *client HELLO* e envia um pacote *server HELLO* para o cliente; o cliente confirma o recebimento enviando um pacote de confirmação para o servidor; o servidor envia o certificado digital ssl para o cliente no formato x509, padrão para certificados ssl, e o cliente envia a confirmação do recebimento do pacote. Feito todo o processo de troca de informações, ambas as partes negociam a chave que realizará a criptografia e descriptografia dos dados, é feito o envio de uma amostra de mensagem criptografada para verificar se as chaves de ambos os lados consegue descriptografar a mensagem, e por fim a feito a troca das mensagens criptografadas entre cliente e servidor.

O TLS pode ser utilizado para acrescentar segurança a outros protocolos, oferecendo uma camada de segurança aos serviços originalmente inseguros, como o HTTP, conforme mostrado na seção seguinte.

2.7 HTTPS

De acordo com Stallings (2014), HTTPS refere-se à combinação do protocolo HTTP sobre o TLS para implementar a comunicação segura entre cliente e servidor. Uma conexão HTTP padrão usa a porta 80 e este protocolo não garante confidencialidade nos dados transmitidos, pois toda a informação é transmitida em texto puro. Se for especificado HTTPS, a porta 443 é usada para invocar o TLS. Ainda segundo Stallings (2014), quando o HTTPS é usado, os seguintes elementos da comunicação são encriptados:

- URL do documento solicitado;
- conteúdo do documento;
- conteúdo dos formulários do navegador (preenchidos pelo usuário do navegador);
- *cookies* enviados do navegador ao servidor e do servidor ao navegador; e
- conteúdo do cabeçalho HTTP.

Documentada na RFC 2818, o uso do HTTP sobre o TLS fornece segurança nos dados trafegados, distinguindo o tráfego seguro do inseguro utilizando uma porta de servidor diferente, podendo ser utilizado a mesma porta. A RFC 2817 descreve mecanismos que permitem que os serviços compartilhem a mesma porta.

O uso do HTTPS demonstra-se mais seguro em relação ao HTTP, por transmitir informações encriptadas e confiabilidade do serviço que utiliza. Neste trabalho, sua implementação será dada como uma alternativa de segurança para evitar a fragilidade quando se utiliza apenas o HTTP.

2.8 IDS / IPS

Em Segurança da Informação o termo IDS e IPS são bastante empregadas quando se trata de segurança. O IDS (*Intrusion Detection System*) é um tipo de solução para automatizar a detecção de acessos não autorizados em redes de computadores. Assim, os IPS têm como objetivo proteger a rede contra as ameaças, auxiliando na detecção e prevenção de ataques, analisando os fluxos de dados trafegados na rede.

Documentado na RFC 4766, um IDS é composto pelos seguintes componentes:

- sensor: realiza a coleta dos dados e encaminha os dados para o analisador, dados coletados são armazenados em arquivos *logs*, sendo o local que armazena os registros de eventos em um sistema;
- analisador: analisa os dados coletados pelo sensor e verifica atividades não autorizadas e eventos que possa ser do interesse do administrador de segurança;
- gerente: componente responsável por gerenciar o processo e o funcionamento da rede, normalmente esse gerenciamento inclui a configuração do sensor, analisador, notificação de evento e relatórios.

De acordo com a SEGINFO (2010), existem diferentes tipos de IDS, podendo ser aplicados em diferentes contexto, como aqueles baseados em assinaturas, anomalias, monitoramento de rede, monitoramento de *hosts* etc. A detecção por assinaturas analisa eventos baseados em padrões ou atividades maliciosas, que são pré-definidas a partir de ataques conhecidos. A detecção baseada em anomalias analisa o comportamento rotineiro do usuário e, caso essa rotina seja fora do normal, emite um alarme ao administrador, sendo que algumas vezes são emitidos alarmes falsos, não representando um ataque ou ação maliciosa. Detecção baseada no monitoramento de rede, mais conhecido como NIDS (*Network Intrusion Detection System*), monitora o tráfego da rede a fim de detectar atividades maliciosas como *scanner* de rede, monitorando o conteúdo e o tráfego dos dados na rede. Os sistemas baseados em *host*, conhecido como HIDS (*Host Intrusion Detection System*), tem como objetivo analisar *logs* e eventos, verificando permissões e possíveis alterações, sendo utilizados mais em servidores.

De acordo com o Portal GSTI (2017), Sistema de Prevenção de Intrusão é conhecido com IPS, o IPS trabalha com a prevenção de ameaças através da análise dos fluxos de tráfego de rede, sendo uma solução que atua no combate de ameaças em ambientes de redes, sendo um complemento ao IDS. O IPS é uma ferramenta com a capacidade de analisar, identificar e prevenir um ataque, bloqueando determinado evento, fortalecendo a técnica de detecção de intrusão, tornando a rede mais segura. De acordo com a OGASEC (2018), os IPSs oferecem vários benefícios para uma rede, benefícios que podem ser:

- detecta e interrompe ataques que outros controles de segurança não o fazem;
- suporta a personalização de recursos de detecção para interromper atividades que são de interesse apenas para uma única organização;
- reduz a quantidade de tráfego de rede que atinge outros controles de segurança, o que reduz a carga de trabalho para esses controles e os protege contra ataques diretos.

Essas soluções se tornam importantes para a segurança de uma rede quando implementados corretamente. Existem ferramentas de segurança que possibilitam o uso do IDS com o IPS, como o *Snort* que é uma ferramenta *open source*.

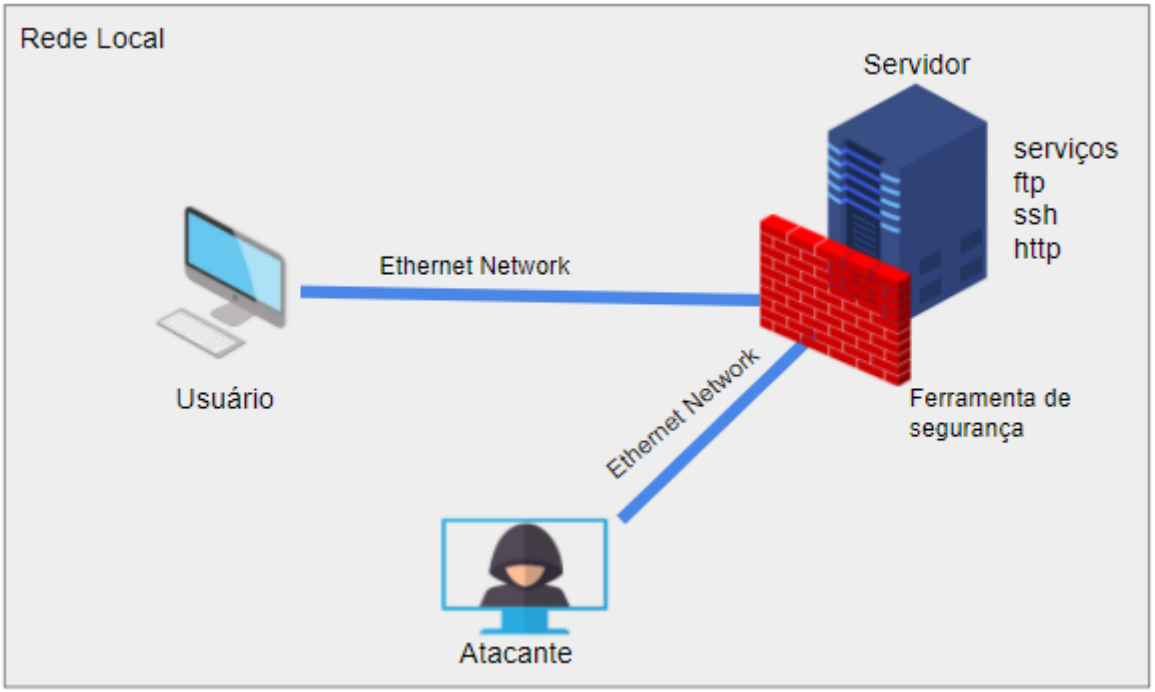
3 MATERIAIS E MÉTODOS

Nesta seção são apresentados os materiais e os métodos utilizados para a realização do trabalho proposto, que consiste na configuração do ambiente de rede para a implementação e demonstração de uma camada de segurança para os serviços ftp, http e ssh em um servidor, bem como a descrição dos ataques de força bruta e interceptação de dados, com o intuito de atingir os objetivos propostos.

3.1 AMBIENTE DE REDE

No trabalho desenvolvido no estágio por GOMES (2020), foi configurado o ambiente de rede, com um servidor vulnerável sobre o qual foram realizados ataques para demonstrar suas fragilidades. Nesse contexto, os ataques foram bem sucedidos, demonstrando vulnerabilidades nos serviços e na comunicação entre cliente e servidor. Neste Trabalho, o ambiente será replicado e serão configuradas ferramentas de segurança para amenizar as vulnerabilidades encontradas. A Figura 5 apresenta a estrutura do ambiente que será criado. Sobre os serviços instalados no servidor da rede, serão realizados ataques para verificar a segurança após a configuração das ferramentas de segurança.

Figura 5 - Arquitetura da rede



Para criação da rede, será utilizada a máquina física e, nela, serão configuradas duas máquinas virtualizadas com o VirtualBox da Oracle, conectadas a uma rede local através de cabo *ethernet*. A configuração de rede nas máquinas virtuais será em modo *bridge*, assim, se

conectarão à rede diretamente, obtendo o endereço IP da rede local. As máquinas configuradas no ambiente serão:

- Servidor: serão realizadas configurações dos serviços de rede ProFTPD (FTP), Apache (HTTP) e OpenSSH (SSH), bem como das ferramentas de segurança, com a finalidade de demonstrar a eficiência das ferramentas em proteger a máquina de ataques, sendo utilizado uma máquina virtual;
- Máquina do usuário: será utilizada para que o cliente solicite os serviços de rede, de forma a gerar tráfego na rede e se tornar alvo dos ataques. Será utilizado o sistema operacional Windows 10, sendo utilizado uma máquina física;
- Máquina do Atacante: será utilizado o sistema operacional Kali Linux e as ferramentas Nmap, Metasploit e Wireshark, para a realização dos testes de vulnerabilidade na rede,sendo utilizado uma máquina virtual.

3.2 SOFTWARES

Esta seção apresenta os softwares que serão utilizados para a criação do ambiente, realização dos ataques e proteção da rede.

3.2.1 Sistemas Operacionais

Para a máquina atacante será utilizado o sistema operacional Kali Linux, que oferece um conjunto de ferramentas instaladas por padrão para a realização dos ataques. Tais ferramentas podem ser instaladas e configuradas em outros sistemas operacionais, mas demandaria mais trabalho.

Na máquina do usuário será usado o sistema operacional Windows que é comumente utilizado na maioria das empresas, e na máquina do servidor, será utilizado o sistema operacional Ubuntu que é uma distribuição Linux.

3.2.2 Serviços de Rede

Os serviços de redes que serão configurados no servidor e sofreram os ataques são:

- ProFTPD: *software Open Source* (código aberto) que implementa o protocolo FTP (*File Transfer Protocol*), que permite a transferência de arquivos entre uma máquina e outra pela rede. Para Chung (2014), é o protocolo mais utilizado e conhecido para baixar arquivos da Internet. Este serviço se torna inseguro quando não é utilizado mecanismos e ferramentas de segurança, pois o protocolo apresenta insegurança na comunicação dos dados trafegados;

- Apache HTTP *Server*: *software Open Source* (código aberto) que implementa o protocolo HTTP, que é um protocolo de comunicação entre sistemas de informação de hipermídia, distribuídos e colaborativos, sendo a base de comunicação com a rede mundial de computadores, a WWW (*World Wide Web*), possibilitando a transferência de dados na internet. De acordo com KUROSE (2006), seu funcionamento acontece quando um cliente solicita uma página web e o servidor retorna esta página por meio do protocolo HTTP para o cliente;
- OpenSSH: ferramenta que implementa o protocolo SSH, que permite acessar outras máquinas que estejam conectadas à rede, tal acesso permite gerenciar e modificar de forma remota outra máquina. O protocolo, por padrão, utiliza mecanismos de criptografia na comunicação dos dados, assim, torna-se difícil visualizar o conteúdo por terceiros que porventura venham capturar os dados em uma rede insegura. Métodos de força bruta a esses serviços são bastante utilizados, podendo dar acesso a pessoas não autorizadas, serviços configurados de forma inadequada permite a realização desse ataque, deixando-a vulnerável.

3.2.3 Ataques utilizados

Essa seção apresenta os ataques realizados por GOMES(2020) no ambiente de rede sem o uso de ferramentas de segurança, no qual foram realizados os ataques de força bruta e a interceptação de dados, que serão descritos a seguir:

- Força bruta: o ataque consiste em uma abordagem de tentativa e erro a um determinado usuário e senha de um serviço, podendo ser utilizado listas de possíveis usuários e senhas para a realização do ataque. Esse ataque é um método de ataque antigo, mas bastante popular e utilizado entre os hackers. De acordo com a CERT.BR (2017) aponta alguns problemas que podem ser causados por este tipo de ataque, tais como: perda de acesso; executar ações maliciosas, como apagar arquivos, obter informações confidenciais e instalar códigos maliciosos; bloqueio de contas.
- Interceptação de tráfego (*Sniffing*): técnica que consiste em inspecionar os dados trafegados em uma rede, utilizando programas específicos conhecidos como *sniffers*. Programas que possibilitam analisar o conteúdo de um “pacote” conhecido como datagrama que por sua vez consiste em um conjunto de informações de controle e dados do usuário. Esta técnica pode ser utilizada de forma:

> "Legítima: por administradores de redes, para detectar problemas, analisar desempenho e monitorar atividades maliciosas relativas aos computadores ou redes por eles administrados.
> Maliciosa: por atacantes, para capturar informações sensíveis, como senhas, números de cartão de crédito e o conteúdo de arquivos confidenciais que estejam trafegando por meio de conexões inseguras, ou seja, sem criptografia." CERT.BR (2017).

Para um atacante, o uso dessa técnica se torna bastante útil quando se encontra uma rede insegura, coletando informações que poderão ser utilizadas posteriormente para uma ação maliciosa. No trabalho, a interceptação de tráfego será utilizado nos serviços Apache e ProFTPD.

Os ataques descritos serão replicados em um ambiente no qual possuirá mecanismos e ferramentas de segurança implementados. As ferramentas de ataque utilizadas para a realização dos ataques, serão descritas na seção seguinte.

3.2.4 Ferramentas de Ataque

Para a realização dos ataques na rede, foram utilizadas as ferramentas Nmap, Metasploit e Wireshark. Ferramentas que são descritas a seguir:

- Nmap (Network Mapper – Mapeador de Redes): é uma ferramenta *Open Source* (código aberto), que contém uma grande variedade de recursos e funcionalidades, como mapeamento de rede, detecção de *hardware* de dispositivos da rede, serviços ativos e entre outras. Segundo Bezerra (2012), o Nmap é uma das ferramentas de análise de rede mais conhecidas entre os profissionais de rede e entre criminosos por ser livre e eficiente quanto a sua utilização. No trabalho, a ferramenta terá como objetivo realizar a varredura de portas e detecção dos serviços na rede, para, a partir dessa informação, a encontrar vulnerabilidades nesses serviços;
- Wireshark: ferramenta *Open Source* (código aberto) que suporta dezenas de protocolos e sendo popularmente utilizada por especialistas na análise e captura de pacotes em uma rede. De acordo com Bullock (2017), a ferramenta permite entender os dados capturados em uma rede, podendo detalhar o conteúdo de cada pacote;
- Metasploit Framework: ferramenta de código aberto que disponibiliza um conjunto de *exploits*, *payloads* e *auxiliary* que ajudam na realização dos testes de intrusão em uma rede.

Os *Exploits* são códigos maliciosos capazes de aproveitar uma determinada vulnerabilidade no sistema, de acordo com a Kaspersky (2016). De acordo com a empresa Hacker Security (2018), *payloads* podem ser definidos como uma sequência de passos para atingir um objetivo, refere-se à parte de um código malicioso que executa uma ação nociva no sistema do alvo. De acordo com a Offensive Security (2007), *auxiliary* são módulos que permite realizar varreduras, realizar teste de software, analisar a rede e entre outras. Esses módulos são bastante utilizados e auxiliam na realização de um ataque.

3.2.5 Ferramentas de segurança implantada

As ferramentas de segurança utilizadas para amenizar as fragilidades nos serviços de redes foram o OpenSSL e a ferramenta IDS/IPS Snort que são descritos a seguir:

- OpenSSL: ferramenta para implementar mecanismos e protocolos de segurança, baseado no projeto SSLeay. De acordo com a IBM (2020) a ferramenta **openssl** pode ser utilizada para fins de gerenciamento de certificados. Com isso, o openssl permite gerar vários formatos de certificado, sendo utilizado neste trabalho o formato padrão **X.509** para a geração de certificado digital para a *web*. Russell (2019). Com base nestas informações, a ferramenta OpenSSL foi utilizada para gerar o certificado digital associado aos serviços Apache e ProFTPD.
- Ferramenta IDS/IPS Snort: ferramenta de código aberto, disponibilizado para *download* no site snort.org. A ferramenta pode ser definida como:

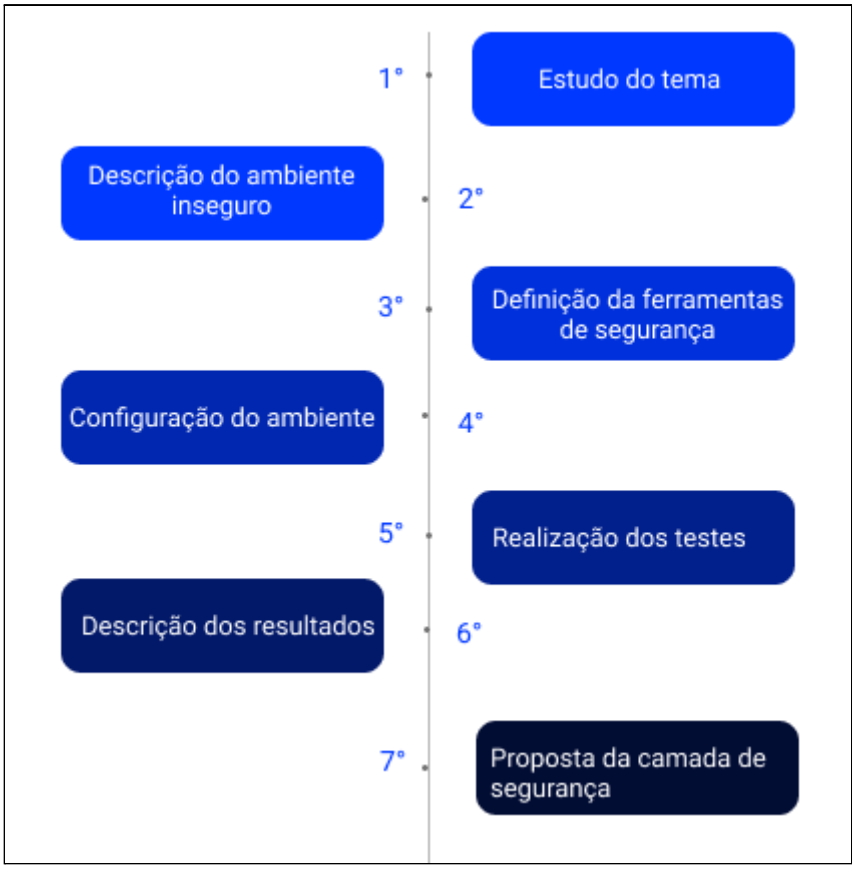
> "Sistema de Prevenção de Intrusão (IPS) de código aberto mais importante do mundo. O Snort IPS usa uma série de regras que ajudam a definir a atividade de rede mal-intencionada e usa essas regras para encontrar pacotes que correspondam a eles e gerar alertas para os usuários." Snort.org (2021?).

A ferramenta Snort possibilita a detecção e bloqueio de ameaças internas e externas na rede, sendo utilizado para evitar o ataque de força bruta , bem como a detecção e bloqueio de *scanners* de rede.

3.3 METODOLOGIA

Nessa seção são apresentadas as etapas executadas no processo de implementação de uma camada de segurança em um servidor de rede, conforme a Figura 6.

Figura 6 - Fluxo da Metodologia



A primeira etapa deste projeto consistiu em realizar o estudo bibliográfico sobre os conceitos relacionados ao tema do trabalho, que foram: segurança da informação, protocolo ftp, protocolo ssl, https, ids e ips.

Já na segunda etapa foi replicado e descrito o ambiente inseguro, apresentado no trabalho de Gomes (2020), no qual foram instalados os serviços Apache (HTTP), ProFTPD (FTP) e OpenSSH (SSH). Esses serviços por *default* apresentam fragilidades que, quando exploradas, podem levar a problemas relacionados à segurança, desde vazamento de dados confidenciais a danos financeiros.

Na terceira etapa foram definidas as ferramentas de segurança que foram implementadas para minimizar os problemas de vulnerabilidade dos serviços, apresentados no trabalho anterior, adicionando uma camada de segurança à rede insegura. A ferramenta de segurança Snort foi escolhida para prevenir o ataque de força bruta no serviço OpenSSH e analisar atividades suspeitas sobre a rede, como um *scanner* de rede. Para os serviços Apache e ProFTPD foram realizados a criptografia dos dados enviados, utilizando o certificado SSL/TLS sobre os protocolos HTTP e FTP, na finalidade de proteger as informações contidas no envio dos dados trafegados por estes serviços. Essas ferramentas foram escolhidas

considerando que, normalmente, são apontadas como as soluções de segurança muito usadas para os serviços e plataformas implantados no trabalho, não tendo sido realizada e descrita uma análise aprofundada sobre elas antes da definição.

A quarta etapa envolveu a implantação das ferramentas de segurança definidas na etapa anterior, criando a camada de segurança para proteger os serviços de rede configurados.

Com o ambiente configurado, a quinta etapa consistiu na realização dos testes sobre os serviços Apache (HTTP), ProFTPD (FTP) e OpenSSH (SSH). Onde foram realizados os mesmos ataques que são apresentados no trabalho de Gomes (2020), do qual o ambiente inseguro foi replicado. O objetivo foi demonstrar que, após a instalação das ferramentas de segurança, as vulnerabilidades do ambiente inseguro foram mitigadas.

Na sexta etapa foram descritos os resultados dos testes, apresentando os ataques realizados e o comportamento dos mecanismos de segurança implantados em relação a eles, no qual foi possível mitigar as vulnerabilidades existentes anteriormente.

Por fim, na sétima etapa do trabalho foi proposta uma solução para mitigar as vulnerabilidades nos serviços de rede, adicionando uma camada de segurança que ameniza a possibilidade de ataques.

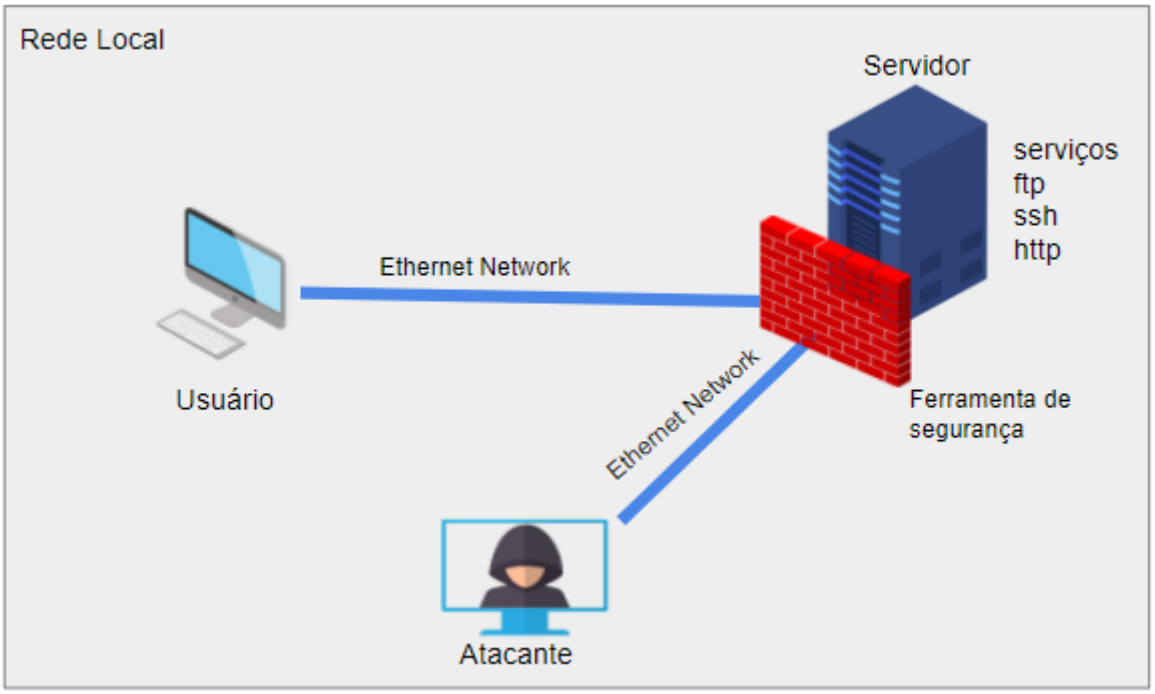
4 RESULTADOS

Nesta seção é descrito o processo de execução e resultados dos testes de cada etapa da solução proposta, que consistiu em implementar mecanismos de segurança em um servidor inseguro para propor uma configuração de uma camada de segurança em um servidor de rede. A seção a seguir traz consigo o ambiente inseguro no qual foram implementados os mecanismos de segurança a fim de mitigar as vulnerabilidades existentes nesse ambiente.

4.1 AMBIENTE DE REDE INSEGURO

No trabalho desenvolvido no estágio por Gomes (2020) foi configurado o ambiente de rede, com um servidor vulnerável, sobre o qual foram realizados ataques para demonstrar suas fragilidades. Sobre esse servidor, foram realizados ataques bem sucedidos, demonstrando vulnerabilidades nos serviços e na comunicação entre cliente e servidor. Neste Trabalho, o ambiente foi replicado e foram configurados mecanismos de segurança sobre os serviços inseguros, tais mecanismos foram implantados nos serviços de rede para oferecer uma maior segurança e mitigar danos causados por um ataque que porventura venha a ocorrer.

Figura 7 - Estrutura do ambiente de rede



O ambiente apresentado na figura 7 é composto por uma máquina física e, nela, foram configuradas duas máquinas virtualizadas com o VirtualBox da Oracle, conectadas a uma rede local através de cabo *ethernet*. A configuração de rede nas máquinas virtuais foi em modo *bridge*, assim, se conectaram à rede diretamente, obtendo o endereço IP da rede local. No ambiente, os serviços configurados foram o Apache, OpenSSH e ProFTPD.

No trabalho de Gomes (2020), foram realizados ataques a partir dos quais foram encontradas as seguintes vulnerabilidades:

- Port scan: utilizando a ferramenta Nmap foi possível, através de um *scanner* de rede, descobrir versões, portas abertas e serviços existentes no servidor. A realização de um *port scan* (varredura) na rede é um método empregado pelos criminosos para identificar pontos fracos e possíveis brechas que podem ser exploradas a fim de forçar a entrada em sistemas operacionais;
- Interceptação de tráfego (*Sniffing*): utilizando a ferramenta Wireshark foi possível, através de uma captura de dados na rede, obter o *login* e *password* de um usuário da rede no processo de conexão aos serviços Apache e ProFTPD, em que essas informações foram trafegadas em texto plano;
- Força bruta: ataque realizado utilizando a ferramenta Metasploit, esse ataque apresentou sucesso ao realizar diversas tentativas de acesso ao serviço de acesso remoto OpenSSH, que não foram detectadas. Quando não se implementa mecanismos

de segurança que limitam a tentativa de acesso a uma determinada conexão, esse tipo de ataque pode ter êxito em um determinado momento.

A seção a seguir apresenta a descrição e a implementação da camada de segurança que foi realizada sobre o ambiente inseguro.

4.1.1 Camada de segurança implementada

Esta seção apresenta a descrição dos problemas e as soluções encontradas para implantar a camada de segurança nos serviços do servidor. A camada de segurança consiste em adicionar ao servidor as ferramentas de segurança para mitigar as vulnerabilidades detectadas nos serviços de rede.

Os problemas encontrados nos serviços de rede foram a captura de texto plano nos serviços ProFTPD e HTTP, a não detecção da realização da varredura na rede e o sucesso do ataque de força bruta realizado sobre o OpenSSH. A seguir, são apresentadas cada uma das vulnerabilidades identificadas nos serviços do servidor e a solução de segurança implementada para resolver esse problema.

4.2 ADICIONANDO SEGURANÇA AO PROFTPD E AO APACHE HTTP

Os serviços ProFTPD e Apache HTTP apresentam, nos protocolos que as definem, a fragilidade de ser possível ler todo o conteúdo em texto plano na comunicação entre cliente e servidor. A fim de mitigar essa vulnerabilidade, a solução proposta para garantir uma camada de segurança aos serviços ProFTPD e HTTP Apache foi acrescentar criptografia aos serviços e usar um certificado digital OpenSSL.

4.2.1 Criando o certificado digital - OpenSSL

O problema encontrado nos serviços ProFTPD e Apache HTTP foi que as informações transmitidas foram capturadas em texto plano. Para isso, a solução encontrada para os dois serviços foi a utilização do OpenSSL para adicionar criptografia aos serviços por meio do certificado digital SSL/TLS. De acordo com a Kaspersky (2021), o certificado ssl é um certificado digital que autentica a identidade de um site e possibilita uma conexão criptografada utilizando o protocolo SSL/TLS, esse protocolo de segurança cria a

comunicação criptografada entre cliente e servidor, garantindo aos usuários a autenticidade e segurança para o compartilhamento de informações. Um certificado contém uma chave pública e uma chave privada associada, informações sobre o proprietário do certificado e a assinatura do proprietário (IBM, 2020).

Neste trabalho, por ser um ambiente de teste, foi utilizado o certificado autoassinado, que não verifica a origem do certificado por meio de uma autoridade de certificação de terceiros. É aconselhável utilizar esse tipo de certificado apenas em ambientes internos, no qual é possível estabelecer uma conexão segura sem a necessidade de acesso a internet para seu uso, sendo bastante utilizado para testar uma configuração de SSL antes de criar e instalar um certificado assinado fornecido por uma autoridade de certificação (AC). Já para redes com acessos externos, não é aconselhável a sua utilização, sendo importante usar os certificados assinados por autoridades de certificação (ACs), que é uma entidade confiável e responsável pela emissão dos certificados digitais.

A Figura 8 apresenta o comando que foi utilizado para gerar os certificados para o serviço ProFTPD e Apache HTTP, via *prompt* de comando.

Figura 8 - comando para gerar o certificado digital

openssl req -x509 -nodes -newkey rsa: 2048 -keyout caminho/chave **-out** caminho/certificado **-days 365**

O comando apresentado na figura 8 gera o certificado e uma nova chave privada que utiliza o algoritmo RSA com tamanho de 2048 bits, válido por 365 dias. Na execução desse comando, são solicitadas diversas informações como o nome, estado, localidade, organização, nome do servidor e *e-mail* para incorporar outros detalhes ao certificado. A seção a seguir apresenta o processo de implementação da criptografia nos serviços ProFTPD e Apache HTTP usando o certificado criado, bem como os testes em cada serviço.

4.2.2 Configuração da Criptografia no ProFTPD

Para a implementação da criptografia no serviço ProFTPD foi necessário modificar os arquivos de configuração ***tls.conf*** e ***proftpd.conf***, localizados no diretório /***etc/proftpd/***. A figura 9 apresenta o arquivo *tls.conf,* modificado.

Figura 9 - arquivo tls.conf

```
TLSEngine                on
TLSLog                   /var/log/proftpd/tls.log
TLSProtocol              SSLv23
TLSRSACertificateFile    /etc/proftpd/ssl/proftpdCertificate.pem
TLSRSACertificateKeyFile /etc/proftpd/ssl/proftpdServerkey.pem
TLSRequired              on
```

Nas três primeiras linhas foi realizada a ativação de conexões TLS/SSL, armazenamento de logs e definição do protocolo. Essa informação faz com que o cliente possa solicitar a criptografia no serviço e visualizar os registros de eventos para conhecer o seu comportamento no passado. Nas três últimas linhas, foi realizada a associação do certificado ao serviço, configurando o caminho da chave privada e do certificado.

O arquivo *proftpd.conf* foi modificado para incluir as configurações do *tls.conf*, sendo adicionada uma linha com a instrução apresentada na figura 10.

Figura 10 - arquivo proftpd.conf

```
include /etc/proftpd/tls.conf
```

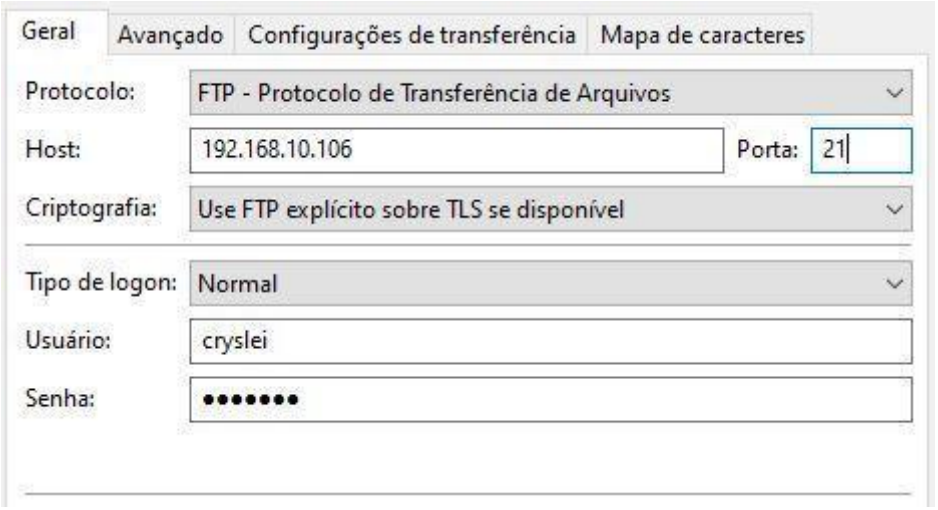
Após o procedimento da implementação da criptografia, com a associação do certificado, foi necessário reiniciar o serviço para que as alterações fossem efetivadas. Para verificar a segurança da rede, após essa etapa, foi utilizada a ferramenta FileZilla para realizar a comunicação entre cliente e servidor e a ferramenta wireshark para capturar os dados trafegados nessa comunicação.

4.2.2.1 Resultados dos Testes

Para o acesso ao serviço ProFTPD, a conexão do cliente com o servidor foi realizada com a ferramenta FileZilla, que é um cliente FTP, com o intuito de gerar tráfego na rede para a realização dos testes. A Figura 11 apresenta as informações acrescentadas ao Filezilla para conectar-se ao servidor no *host* 192.168.10.106. No campo **Criptografia**, a opção escolhida

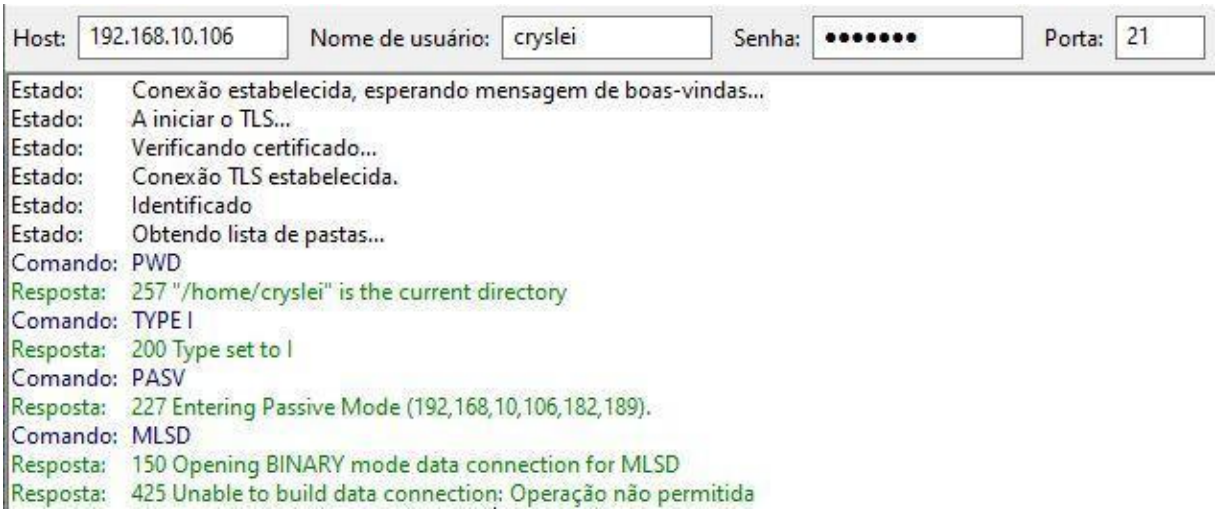
foi utilizar a comunicação segura por meio SSL/TLS, que é a utilização do certificado digital ssl sobre o serviço, sendo que essa opção só tem efeito pelo fato de que a criptografia foi implementada no serviço FTP.

Figura 11 - Acesso ao servidor ftp via interface gráfica



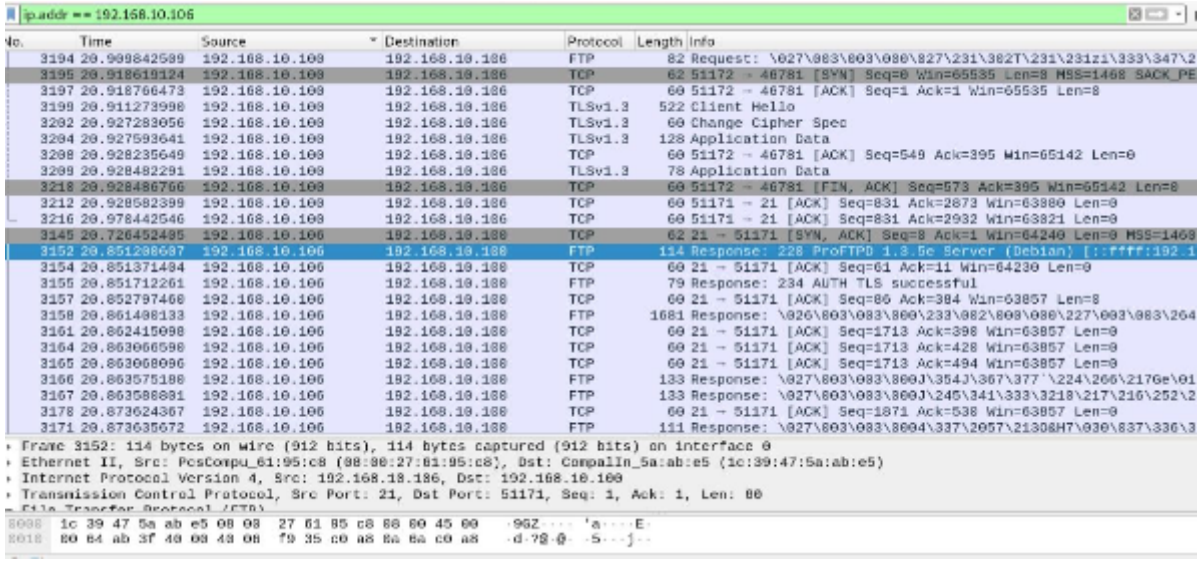
A figura 12 apresenta as informações da conexão segura estabelecida. As linhas 2 a 5 demonstram que foi realizada uma conexão criptografada por meio do TLS.

Figura 12 - Conexão ao serviço FTP sobre TLS



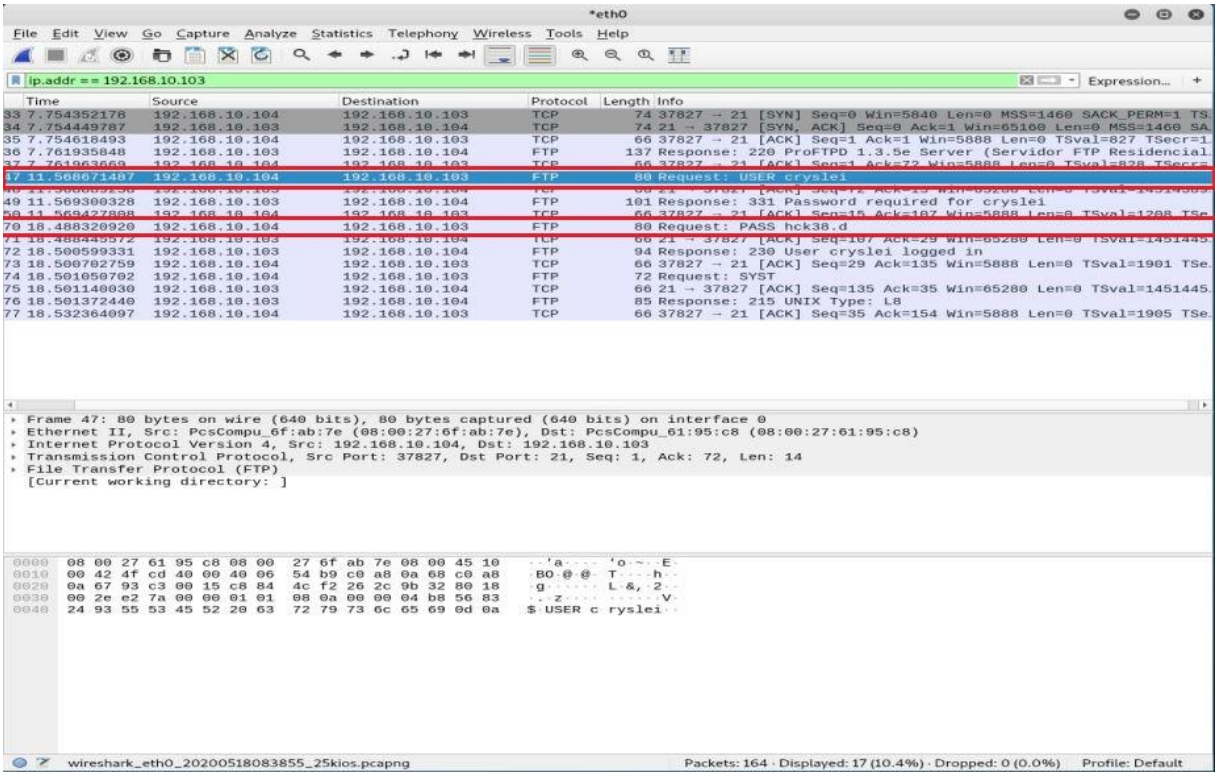
Após estabelecida a conexão do cliente ao servidor, na máquina do atacante, a ferramenta *Wireshark* realizou a captura dos dados de conexão trafegados na rede. A Figura 13 apresenta o retorno do *Wireshark*.

Figura 13 - Captura dos dados com endereço ao servidor FTP



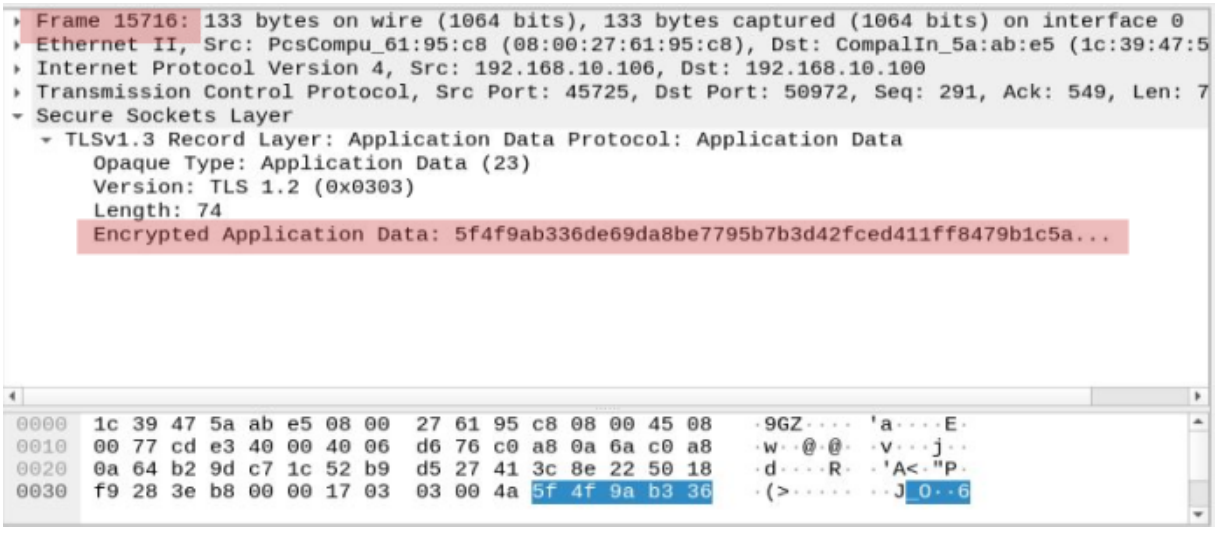
Após feita a captura dos dados com a ferramenta Wireshark, foi utilizado o filtro **ip.addr == 192.168.10.106**, que é o IP da máquina servidor, para apresentar apenas os dados da conexão do teste. Por gerar muitos dados referente ao teste, desde o estabelecimento da conexão à troca de informações, foi realizada a análise a partir do pacote *Application Data*, sendo que, os dados gerados a partir desse ponto são referentes aos dados da aplicação. As figuras 14 e 15 apresentam, respectivamente, os pacotes capturados em texto plano, antes de configurar a criptografia, e os pacotes criptografados, capturados após a configuração da segurança.

Figura 14 - Conteúdo capturado em texto plano



A figura 14 apresenta os pacotes capturados em texto plano, sendo possível visualizar o conteúdo do pacote. No campo info é mencionado o nome do usuário, USER **cryslei**. E, no pacote 70, é apresentada a senha do usuário, PASS **hck38.d**.

Figura 15 - Conteúdo do pacote criptografado



Feita a análise do conteúdo *Application Data* do pacote 15716, pode-se observar que a comunicação foi criptografada em *Encrypted Application Data*, segunda linha destacada na figura 15, não sendo possível visualizar o conteúdo da comunicação, de forma que o objetivo de adicionar uma camada de segurança ao serviço ProFTPD foi atingido.

4.2.3 Configuração da Criptografia no Apache HTTP

Para que fosse adicionada a criptografia ao serviço foi preciso adicionar as linhas no arquivo *default-ssl.conf*, conforme apresentado na figura 15. Indicando que o certificado ssl foi implementado no serviço Apache HTTP.

.

Figura 16 - arquivo default-ssl.conf

```
SSLCertificateFile /etc/ssl/certs/certificado.crt
SSLCertificateKeyFile /etc/ssl/private/certificado.key
```

A figura 16 apresenta os campos que foram modificados no arquivo *default-ssl.conf*, para a associação do certificado ao serviço, passando o caminho da chave privada e o certificado. A figura 17 apresenta a linha adicionada no arquivo *apache2.conf* para implementar as modificações no serviço.

Figura 17 - arquivo apache2.conf

```
IncludeOptional sites-available/default-ssl.conf
```

Para incluir as configurações do arquivo *default-ssl.conf* ao serviço, foi adicionada a configuração apresentada na Figura 17 no arquivo *etc/apache2/apache2.conf* e, na sequência, foi feita a reinicialização do serviço para que as modificações fossem efetivadas. Para validar a camada de segurança do serviço Apache, foi realizada a captura dos dados trafegados na rede com o uso da ferramenta Wireshark como descrito na próxima sessão.

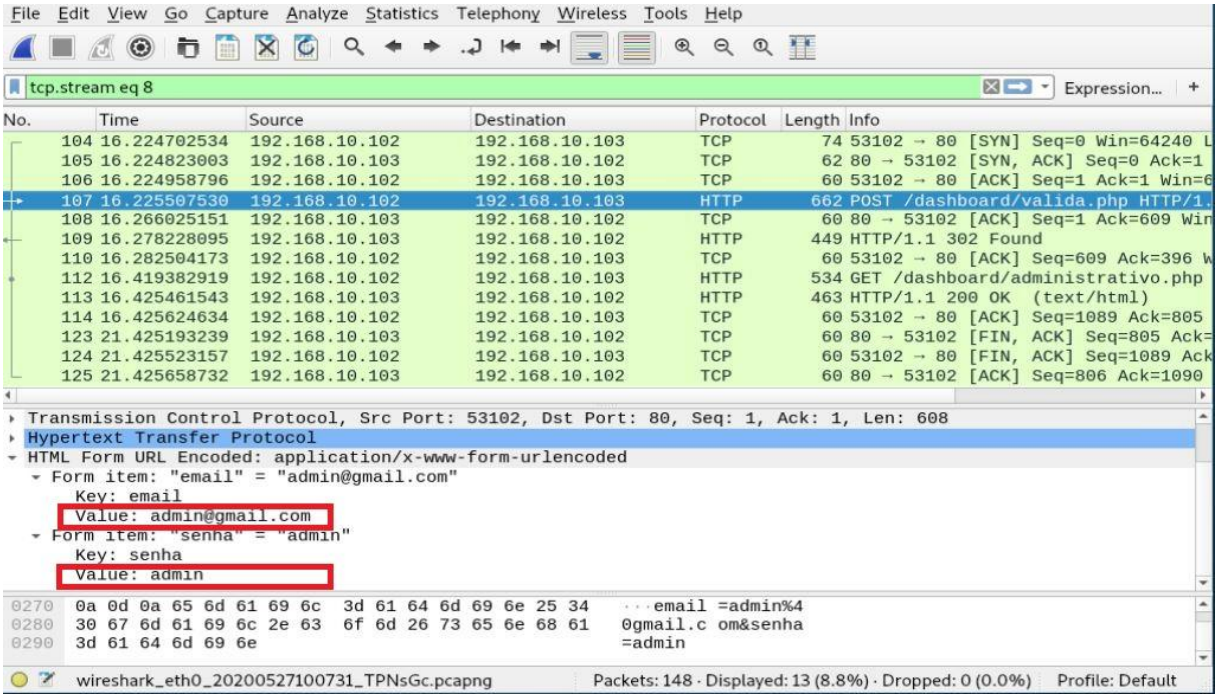
4.2.3.1 Resultados dos Testes

Para realização do teste, foi utilizada a ferramenta Wireshark para a captura dos dados trafegados entre cliente e servidor do serviço Apache. A figura 18 apresenta uma aplicação cliente simples retirada da plataforma de ensino Celke, adaptada e implementada no trabalho, que oferece uma tela de login, apenas para acessar o servidor web e realizar os testes. A tela de login foi disponibilizada no endereço 192.168.10.106 do servidor, sendo acessada por meio do navegador no ambiente interno.

Figura 18 - Acesso ao servidor web

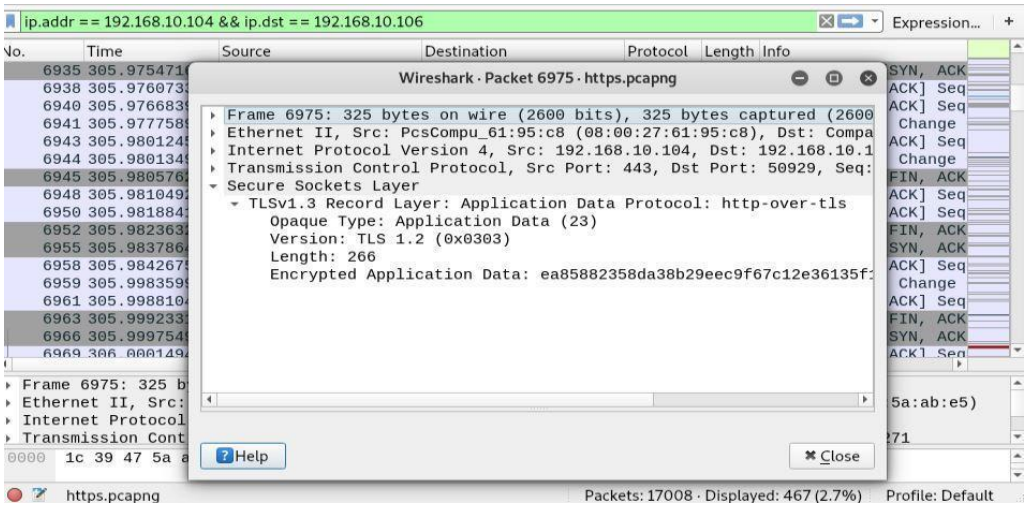
Na troca das informações de conexão, foi gerado tráfego de conteúdo na rede, que foi capturado pela ferramenta Wireshark e, não sendo possível ler o conteúdo por estar criptografado. As figuras 19 e 20 apresentam, respectivamente, os pacotes capturados em texto plano, antes de configurar a criptografia, e os pacotes criptografados, capturados após a configuração da segurança.

Figura 19 - Pacote capturado em texto plano



A figura 19 apresenta os pacotes capturados em texto plano, sendo possível visualizar o conteúdo do pacote. Em destaque é mencionado o email do usuário, **admin@gmail.com** e a senha **admin**.

Figura 20 - Pacote capturado criptografado



Portanto, foi possível atingir o objetivo proposto ao implementar a criptografia na comunicação dos dados, tornando o serviço mais seguro e solucionando o problema da possibilidade de leitura das informações devido a troca de texto plano por padrão.

A outra vulnerabilidade encontrada na rede foi a possibilidade de realização de scanner de rede, utilizando a ferramenta Nmap, que permite ao atacante coletar informações sobre os serviços existentes. A seção a seguir apresenta uma solução para evitar o scanner de rede no servidor, que é a ferramenta Snort.

4.3 EVITANDO SCANNER NA REDE

Essa seção apresenta uma abordagem encontrada para solucionar o problema de possibilidade de realização de scanner de rede, que pode ser definida como uma varredura na rede para captura de informações para realizar um possível ataque aos serviços encontrados. A solução encontrada foi a utilização da ferramenta IDS/IPS Snort para o bloqueio de scanner na rede, dificultando o processo de coleta de informação por parte de uma pessoa mal intencionada.

4.3.1 Configuração da segurança com SNORT

A instalação da ferramenta foi feita com o comando **sudo apt-get install snort** via prompt de comando. Após a instalação, foi alterado no arquivo *snort.conf* o módulo sfportscan, que tem o intuito de detectar *scanners* de rede. A figura 21 apresenta o trecho alterado.

Figura 21 - arquivo snort.conf

```
# Portscan detection. For more information, see README.sfportscan

preprocessor sfportscan: proto { all } memcap { 10000000 } sense_level { low }
```

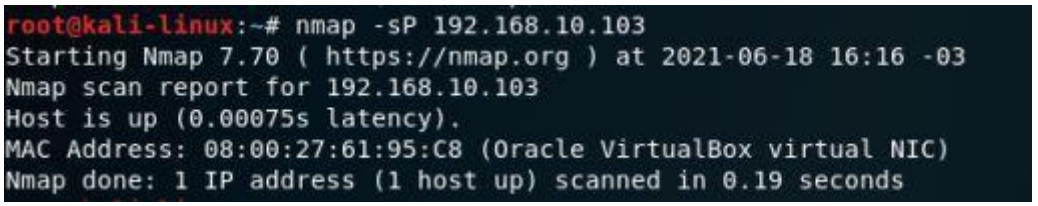
Conforme pode ser visto na figura 21, os parâmetros configurados no módulo *sfportscan* foram: **proto** { all }, para verificar todos os protocolos suportados pela ferramenta; documentado em Snort.org (2004) o parâmetro **memcap** {10000000 }, apresenta o número máximo de bytes a serem alocados para a detecção do portscan; e **sense_level** {low}, que define a sensibilidade da ferramenta quanto a detecção de varredura. Após configurada a ferramenta, foram realizados os testes para verificar se o snort apresentaria mensagens de alerta ou bloquearia as tentativas de *scanner* na rede.

4.3.2 Resultados dos Testes

Para a aplicação do teste na rede, foram utilizadas a ferramenta Nmap, executada na máquina do atacante para fazer a varredura (*scanner*) das portas no servidor, e a ferramenta

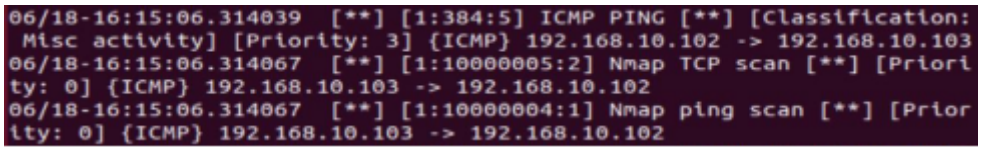
Snort, configurada no servidor para detectar e prevenir a varredura na rede. A figura 22 apresenta o resultado do scanner realizado pelo atacante.

Figura 22 - Scanner com a ferramenta Nmap



Na visão do atacante, a varredura com o Nmap não conseguiu identificar os serviços contidos no servidor, pelo fato de que a ferramenta Snort realizou o bloqueio do scanner na rede. No servidor atacado, a ferramenta Snort gerou o alerta de scanner, como mostra a figura 23.

Figura 23 - Detecção de scanner com a ferramenta Snort



A figura 23 apresenta informações sobre a detecção de *scanner* na rede, que contêm o horário da detecção às 16:15, protocolo utilizado do tipo *Internet Control Message Protocol* (ICMP), classificação *Misc-activity*, prioridade 3 e o endereço de origem 192.168.10.102. No alerta, o Snort classificou a tentativa de ataque como *Misc-activity*, sendo uma atividade de fácil detecção e a prioridade de nível 4, muito baixa. A classificação das prioridades podem ser encontradas em Snort.org (2004), onde o snort atribui níveis de gravidade para cada alerta gerado, que, por padrão, existem 4 tipos de prioridade, (1) alta, (2) média, (3) baixa e (4) muito baixa, sendo possível classificar outros níveis de prioridade de forma customizada no arquivo *classification.config*. Por fim, foi informado que foi realizado o scanner de rede pela máquina atacante 192.168.10.102 contra o servidor 192.168.10.103.

O scanner de rede é classificado como um dos principais métodos utilizados durante a fase de coleta de dados (MITRE, 2021). Dito isso, foi necessário implementar uma camada de segurança para amenizar possíveis ataques a rede relacionadas às informações coletadas por um scanner. Para atingir esse objetivo, a ferramenta Snort apresentou um bom desempenho em detectar e prevenir a coleta de dados pelo atacante.

A ferramenta Snort pode ser utilizada para detecção de ataques de força bruta, ataque bastante utilizado em serviços de rede, e a seção seguinte mostra o uso dessa ferramenta Snort para detectar esse tipo de ataque sobre o serviço OpenSSH.

4.4 ADICIONANDO SEGURANÇA AO OPENSSH

O serviço OpenSSH, por padrão, oferece criptografia na troca de mensagens na rede, sendo que os dados capturados não podem ser lidos em texto plano. Porém, a possibilidade de ocorrer um ataque de força bruta é provável. Para aumentar o grau de segurança ao serviço OpenSSH, foi utilizada a ferramenta Snort para detectar e bloquear o ataque de força bruta, que gera alertas ao administrador para que seja feita uma análise e sejam tomadas as medidas a partir das informações apresentadas no alerta emitido pela ferramenta.

Para que a ferramenta detectasse o ataque de força bruta, foi utilizada a base de regras disponíveis do Snort, que permitem identificar vários tipos de ataques. No site oficial do snort, foi feito o *download* do pacote de regras *Snort v.2.9*, que contém regras criadas pela comunidade e que oferecem alertas que identificam o comportamento de *scanner* de rede.

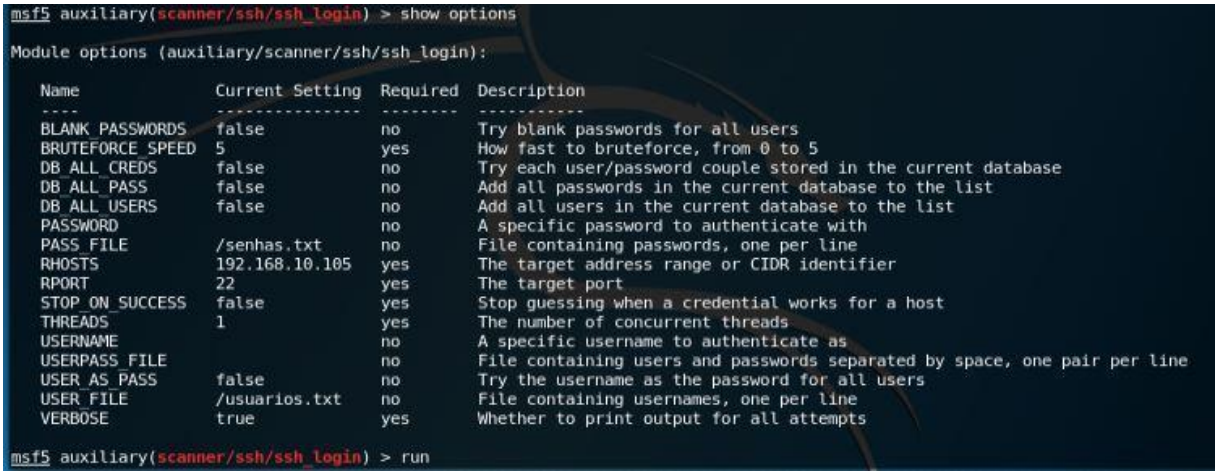
O snort possui três formas de funcionamento: modo captura, que realiza a captura de pacotes trafegados na rede; modo registro, que registra os pacotes que entram na interface de rede; e o modo detecção e prevenção de intrusões: detecta e realiza o bloqueio de intrusos na rede, sendo utilizado no trabalho o modo de detecção e prevenção de intrusões. Para inicializar a ferramenta, armazenar os *logs* e visualizar os alertas no terminal, foi utilizado o comando **sudo snort -d -l /var/log/snort -A console -c /etc/snort/snort.conf**.

Para demonstrar como a ferramenta Snort detecta um ataque de força bruta realizado sobre o serviço OpenSSH, na próxima seção é apresentado o resultado do teste realizado

4.4.1 Resultados dos Testes

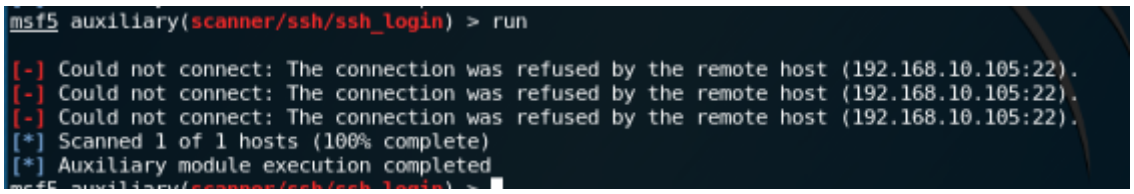
Para realização do teste de força bruta foi utilizada a ferramenta Metasploit, que oferece um conjunto de módulos integrados a ela. Foi utilizado o módulo *auxiliary/scanner/ssh/ssh_login* para disparar o ataque de força sobre o serviço SSH, que realizou sucessivas tentativas de conexão ao servidor OpenSSH. A figura 24 apresenta as configurações que foram realizadas para a realização do ataque.

Figura 24 - Opções preenchidas para executar o módulo ssh_login



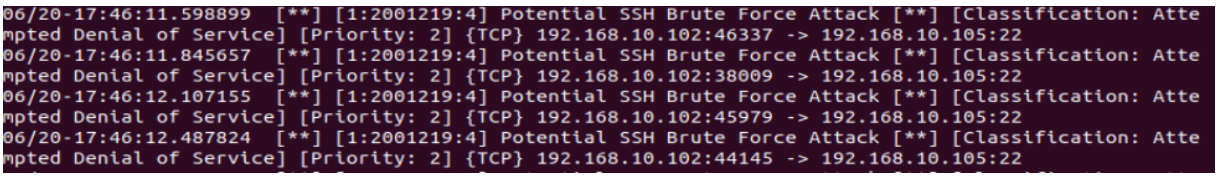
Para a utilização do módulo *scanner/ssh/ssh_login*, foi executado o comando **show options** que retorna as configurações atuais módulo. Onde foi necessário alterar alguns parâmetros do módulo apresentados na figura 24 com o comando **SET [parâmetro] [configuração]**, que foram: **SET RHOST 192.168.10.105** para inserir o *host* do servidor; **SET USER_FILE usuarios.txt** e **SET PASS_FILE senhas.txt** indicando os arquivos que contêm uma lista de possíveis credenciais de usuários e senhas do servidor, arquivos que foram criados para a realizar o teste; **SET VERBOSE true** para possibilitar a visualização das tentativas de acesso ao servidor no terminal. Após a realização da configuração, foi utilizado o comando **run** para a execução do módulo *ssh_login*, como mostra a figura 25.

Figura 25 - Realização do teste de força bruta



A figura 25 apresenta o resultado do teste, apresentando que o ataque não foi bem sucedido, e que, é apresentada apenas a informação que a conexão foi recusada pelo servidor na máquina do atacante. Para a máquina do servidor, esse ataque gerou alertas e o bloqueio do *host* que disparou o ataque de força bruta contra o serviço OpenSSH, como pode ser visto na figura 26.

Figura 26 - Trecho dos alertas gerados pelo Snort



A figura 26 apresenta os alertas gerados pelo SNORT devido a tentativa de ataque de força bruta sobre o serviço SSH, contendo informações como horário, data, endereço de origem e destino, tipo de ataque, prioridade etc.. Assim, foi possível concluir o objetivo proposto de acrescentar uma camada de segurança ao serviço OpenSSH.

A seção a seguir apresenta uma proposta de implementação de uma camada de segurança, a partir das configurações e testes configurados e explicados anteriormente.

4.5 SUGESTÃO DE CAMADA DE SEGURANÇA

Essa seção apresenta uma sugestão de camada de segurança para servidores que oferecem os serviços ProFTPd, Apache HTTP e OpenSSH, visando mitigar as principais vulnerabilidades encontradas nesse tipo de servidor.

A primeira solução de segurança sugerida é adicionar criptografia com OpenSSL e certificado digital aos serviços ProFTPD e Apache HTTP, para resolver o problema da possibilidade do atacante ler os dados que, por padrão, nesses protocolos são trafegados em texto plano. Após a adição da criptografia, se os dados da troca de mensagens forem capturados, o conteúdo não poderá ser lido. Quanto ao certificado digital, é importante realizar a compra de um certificado que seja autenticado por uma autoridade certificadora (AC), que são responsáveis por emitir os certificados digitais, com ele é possível trafegar os dados de forma segura em ambientes internos e externos, por meio da criptografia. É importante que seja autenticado, pois as ACs têm práticas de segurança rigorosas que precisam ser mantidas, o que não acontece com um certificado autoassinado. Cada certificado pode variar de preço de acordo com o tipo, que vão desde certificados simples associados a um domínio, a certificados multidomínio. Em HSI (2021) são apresentados valores que podem variar entre R$ 31,16 a R$1319,70, para certificado simples, e R$100,00 a R$725,00, para certificado multidomínio.

Outra solução sugerida é instalar e configurar a ferramenta Snort para detectar e bloquear anomalias na rede. Com essa ferramenta, foi possível detectar ataques de força bruta e tentativa de *scanner* de rede, sendo realizado os alertas e o bloqueio do endereço de origem. A ferramenta possui diversas funcionalidades que podem aumentar o nível de segurança da

rede, por exemplo, criação de regras e alertas personalizados. O snort pode ser integrado a outras ferramentas que comportam interface gráfica para uma melhor visualização da rede.

Com a ferramenta Snort, os serviços FTP e Apache estarão protegidos contra o ataque de força bruta, caso ocorra uma tentativa de ataque, ele não terá sucesso devido às soluções apresentadas. Além das soluções que foram apresentadas aqui, é importante realizar uma auditoria de segurança para identificar as vulnerabilidades no ambiente, promover senhas complexas, restringir permissões a usuários da rede, utilizar criptografia e manter o sistema atualizado.

5 CONSIDERAÇÕES FINAIS

Este trabalho foi desenvolvido com o objetivo de apresentar uma proposta de camada de segurança para um servidor que disponibiliza os serviços Apache, ProFTPd e OpenSSH, visando auxiliar estudantes e profissionais da área da computação que necessitem garantir segurança em um ambiente de rede ou aprender sobre o tema. No trabalho, foram executadas etapas que envolveram diversas etapas, entre elas, a implantação das ferramentas e mecanismo de segurança no servidor inseguro até a realização dos testes em cada serviço, verificando o comportamento do servidor ao sofrer ataques direcionados a ele.

Para a realização do trabalho, foram estudadas as características dos protocolos e serviços para que fosse possível conhecer as vulnerabilidades e necessidades de garantia de segurança. Com isso, foi possível identificar alguns problemas, tais como: o tráfego de texto plano nos serviços Apache e ProFTPd, possibilidade do ataque de força bruta sobre o serviço OpenSSH e o *scanner* de rede sobre a rede. A partir disso, foram definidas técnicas de pentest, para explorar as vulnerabilidades encontradas e com isso, demonstrar as fragilidades dos serviços.

A solução encontrada para os serviços ProFTPd e Apache *server* foi adicionar criptografia com OpenSSL e o certificado digital, transmitindo os dados na rede de forma criptografada e segura. Foi implementada a ferramenta de segurança Snort no servidor para proteger a rede de um possível ataque de força bruta nos serviços existentes, sendo que, neste trabalho foi realizado o ataque de força bruta somente ao serviço OpenSSH, e por fim, o bloqueio de *scanner* de rede com a ferramenta Snort.

Após a configuração da camada de segurança, foram realizado os testes com as ferramentas Nmap com o propósito de scannear portas abertas no servidor de rede, o Wireshark, para captura e análise de dados na rede e a ferramenta Metasploit, para a realização do ataque de força bruta. Nos testes, foi possível demonstrar que, após a

implementação da camada de segurança, esses ataques não foram bem sucedidos, garantindo assim que as informações estavam seguras em relação a eles.

No desenvolvimento buscou-se demonstrar que a implantação de políticas e práticas de segurança são necessárias para proteger os dados de pessoas mal intencionadas, pois, como foi demonstrado, serviços de rede podem ser vulneráveis a diversos tipos de ataques e existem muitas ferramentas gratuitas de fácil acesso e utilização que possibilitam a realização delas. Além disso, apresentou uma proposta de camada de segurança com o uso de ferramentas gratuitas, de fácil utilização e amplamente documentadas para solucionar as vulnerabilidades dos serviços de rede, sendo que, o único gasto ao se implementar as soluções aqui descritas seria a aquisição de um certificado digital. Um trabalho futuro interessante é adicionar mais serviços ao servidor, testar as vulnerabilidades e propor soluções de segurança aos serviços escolhidos.

REFERÊNCIAS

BEZERRA, A. **“Evitando Hackers”** 2012. Ed Ciência Moderna. 1 ed. Rio de Janeiro.

BRANDINO, Wandreson L. **“Conceitos básicos de internet”**. Universidade Federal do Espírito Santo-UFES. Departamento de Informática. Curso de Pós-Graduação em Redes de Computadores, 1997.

BULLOCK, Jessey; PARKER, Jeff T. **Wireshark for Security Professionals: Using Wireshark and the Metasploit Framework**. John Wiley & Sons, 2017.

CERT.BR. “**Ataques na internet”**. Disponível em:<https://cartilha.cert.br/ataques/>. Acesso em: 01 de Janeiro de 2021.

CELK.COM. **“Sistema de login em PHP”**. Disponível em:<http://celke.com.br/material-gratuito/sistema-de-login-em-php>. Acesso em: 01 de Janeiro de 2021.

CHUNG, Conrad. **“An Introduction to FTP”** 2014. Disponível em: <https://www.2brightsparks.com/resources/articles/an-introduction-to-ftp.pdf>. Acesso em: 02 de Novembro de 2020.

GOMES, Cryslei F.. **“Demonstração de Vulnerabilidade em Servidores de Rede”**. Centro Universitário Luterano de Palmas, Palmas/TO, 2020.

GSTI. **“O que é um Sistema de Detecção de Intrusos”**. Disponivel em: <https://blog.starti.com.br/ids-ips/#:~:text=A%20principal%20diferen%C3%A7a%20entre%20eles,o%20tr%C3%A1fego%20por%20endere%C3%A7o%20IP>. Acesso em: 02 de Novembro de 2020.

HACKER SECURITY. **“O que é um Payload e por que são usados por hackers?”**. Disponível em:<hackersec.com/o-que-e-um-payload-e-por-que-sao-usados-por-hackers/>. Acesso em: 30 de Outubro de 2020.

HSI. **“CERTIFICADO SSL PREÇO | COMPARE ANTES DE COMPRAR | 2021”.** Disponível em:<https://hospedagem-de-sites.info/rankings/preco-certificado-ssl-onde-comprar/>. Acesso em: 30 de Junho de 2021.

IANA. **“Registro de nome de serviço e número de porta de protocolo de transporte”**. Disponível em: <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>. Acesso em: 12 de Novembro de 2020.

INTERNATIONAL BUSINESS MACHINES CORPORATION. **“OpenSSL”**. Disponível em: <https://www.ibm.com/docs/pt-br/ibm-mq/8.0?topic=nss-openssl>. Acesso em: 20 de maio de 2021.

INTERNATIONAL BUSINESS MACHINES CORPORATION. **“Visão Geral do SSL e de Certificados Digitais”.** Disponível em: <https://www.ibm.com/docs/pt-br/sia?topic=o-overview-ssl-digital-certificates-5>. Acesso em: 22 de junho de 2021.

INTERNATIONAL BUSINESS MACHINES CORPORATION. **“Certificados Autoassinados”.** Disponível em: <https://www.ibm.com/docs/pt-br/sia?topic=osdc-self-signed-certificates-29>. Acesso em: 22 de junho de 2021.

INTERNATIONAL TELECOMMUNICATION UNION. **Recommendation X.800. Security Architecture for Open Systems Interconnection for CCITT Applications.** Geneva, 1991. 46f.

KASPERSKY. **“O que são exploits e por que são tão temidos?”**. Disponível em: <www.kaspersky.com.br/blog/exploits-problem-explanation/6010/>. Acesso em: 02 de Novembro de 2020.

KASPERSKY. **“O que é certificado SSL – definição e explicação”**. Disponível em: <https://www.kaspersky.com.br/resource-center/definitions/what-is-a-ssl-certificate>. Acesso em: 02 de Julho de 2021.

KUROSE, James. F. & ROSS, Keith W. **“Redes de Computadores e a internet: Uma abordagem top-down”**, 3ª Edição, Editora Pearson, São Paulo– SP, 2006.

MDN Web Docs. Mozilla Developer Network. **“Content-Type”**. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Headers/Content-Type>. Acesso em: 29 de Novembro de 2020.

MITRE ATT&CK. **“Gather Victim Network Information”**. Disponível em <https://attack.mitre.org/techniques/T1590/> . Acesso em: 19 de Junho de 2021.

NIST. National Institute of Standards and Technology. **“An Introduction to Computer Security: The NIST Handbook”**. Special Publication 800-12. out 1995.

OFFENSIVE SECURITY. **“Auxiliary module reference”**. Disponível em: <www.offensive-security.com/metasploit-unleashed/auxiliary-module-reference/>. Acesso em: 10 de Novembro de 2020.

OGASEC. **“O que é um Sistema de Prevenção de Intrusão (IPS)?”**. disponivel em: <https://www.ogasec.com/blog-ogasec/2018/6/5/o-que-um-sistema-de-preveno-de-intruso-ips >. Acesso em: 29 de Novembro de 2020.

RFC 959: **FILE TRANSFER PROTOCOL (FTP)**. Disponível em: <http://tools.ietf.org/html/rfc959> . Acesso em: 11 de Novembro de 2020.

RFC 2068: **Hypertext Transfer Protocol - HTTP/1.1**. Disponível em:<http://tools.ietf.org/html/rfc2068> . Acesso em: 11 de Novembro de 2020.

RFC 2817: **Upgrading to TLS Within HTTP/1.1**. Disponível em:<http://tools.ietf.org/html/rfc2817> . Acesso em: 29 de Novembro de 2020.

RFC 2818: **HTTP Over TLS**. Disponível em:<http://tools.ietf.org/html/rfc2818> . Acesso em: 11 de Novembro de 2020.

RFC 4250: **The Secure Shell (SSH) Protocol Assigned Numbers**. Disponível em: <http://tools.ietf.org/html/rfc4250> . Acesso em: 11 de Novembro de 2020.

RFC 4251: **The Secure Shell (SSH) Protocol Architecture**. Disponível em: <http://tools.ietf.org/html/rfc4251> . Acesso em: 11 de Novembro de 2020.

RFC 4256: **Generic Message Exchange Authentication for the Secure Shell Protocol (SSH)**. Disponível em:<http://tools.ietf.org/html/rfc4256> . Acesso em: 11 de Novembro de 2020.

RFC 5226: **Guidelines for Writing an IANA Considerations Section in RFCs**. Disponível em:<https://tools.ietf.org/html/rfc5226> . Acesso em: 08 de Janeiro de 2021.

RFC 6335: **Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry**. Disponível em:<https://tools.ietf.org/html/rfc6335> . Acesso em: 08 de Janeiro de 2021.

RFC 7230: **Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing**. Disponível em:<http://tools.ietf.org/html/rfc7230> . Acesso em: 22 de Novembro de 2020.

RFC 7568: **Deprecating Secure Sockets Layer Version 3.0**. Disponível em:<http://tools.ietf.org/html/rfc7568> . Acesso em: 16 de Novembro de 2020.

RFC 8446: **The Transport Layer Security (TLS) Protocol Version 1.3**. Disponível em:<http://tools.ietf.org/html/rfc8446> . Acesso em: 15 de Novembro de 2020.

RFC 4766: **Intrusion Detection Message Exchange Requirements**. Disponível em:<http://tools.ietf.org/html/rfc4766> . Acesso em: 29 de Novembro de 2020.

RUSSELL, A. **“O que é um certificado X.509?”.** Disponível em : <https://www.ssl.com/pt/faqs/o-que-%C3%A9-um-certificado-x-509/>. Acesso em: 20 de Maio de 2021.

SEGINFO. **“Sistemas de Detecção de Intrusões (IDS – Intrusion Detection Systems) usando unicamente softwares Open Source”**. Disponivel em: <https://seginfo.com.br/2010/06/21/sistemas-de-deteccao-de-intrusoes-ids-intrusion-detection-systems-usando-unicamente-softwares-open-source/#tipos-de-sistemas-de-deteccao-de-intrusao>. Acesso em: 29 de Novembro de 2020.

Snort.org, **“Sid 1-45810”**. Disponível em <https://www.snort.org/rule_docs/1-45810>. Acesso em: 19 de Junho de 2021.

Snort.org, **“sfPortscan”**. Disponível em <https://www.snort.org/faq/readme-sfportscan>. Acesso em: 19 de Junho de 2021.

STALLINGS, W. **“Criptografia e Segurança de Redes: Princípios e Práticas”.** [S.l.]: Pearson, 2014. ISBN 8543005892.

TANENBAUM, Andrew S.. **“Redes de Computadores”**. 4. ed. Rio de Janeiro: Elsevier Brasil, 2003.

TANENBAUM, Andrew S.; WETHERALL, David. **“Redes de Computadores”**. 5. ed. São Paulo: Pearson, 2012.