MONITORAMENTO DE ATAQUE LOIC UTILIZANDO O WIRESHARK

LOIC ATTACK MONITORING USING WIRESHARK

Felipe da Silva Cordeiro, Faculdade de Tecnologia de Americana,
felipe.cordeiro6@fatec.sp.gov.br
Gabriel Marcos dos Santos, Faculdade de Tecnologia de Americana,
gabriel.santos336@fatec.sp.gov.br
Horus de Oliveira, Faculdade de Tecnologia de Americana,
horus.oliveira@fatec.sp.gov.br
Wagner José da Silva, Faculdade de Tecnologia de Americana,
wagner.silva@fatec.sp.gov.br

Resumo

O monitoramento de redes de dados é uma atividade complexa que enfrenta desafios éticos e técnicos. Do ponto de vista ético, o monitoramento de redes pode violar a privacidade dos usuários, pois coleta dados pessoais e sensíveis. Além disso, o uso inadequado desses dados pode resultar em discriminação, assédio ou violações dos direitos individuais. Do ponto de vista técnico, o monitoramento de redes requer recursos substanciais, incluindo hardware e software avançados, o que pode ser dispendioso e complexo. É preciso encontrar o equilíbrio adequado entre a busca por segurança e a proteção da privacidade. Para garantir que o monitoramento de redes seja conduzido de maneira ética e eficaz, é preciso estabelecer políticas claras e garantir a conformidade com regulamentações.

Palavras-chave: Monitoramento de redes de dados, Privacidade, WireShark, Consentimento, Jurisdição.

Abstract

This article provides a technical analysis of the ethical issues related to data network monitoring, a field filled with challenges encompassing technical, operational, and ethical concerns. Data network monitoring faces critical issues such as invasion of privacy due to the collection of personal and sensitive data. Furthermore, there are ethical concerns regarding the improper use of collected data, which can lead to discrimination, harassment, or violations of individual rights. The successful implementation of network monitoring requires substantial technical resources, including advanced hardware and software, which can be a costly and complex endeavor. Additionally, striking the right balance between pursuing security and safeguarding privacy is a constant challenge in this field. These ethical and technical issues, which are becoming increasingly complex with new technologies, make data network monitoring an area of great complexity and demand clear policies and compliance with regulations to ensure that it is conducted in an ethical and effective manner.

Keywords: Data Network Monitoring, Privacy, Wireshark, Consent, Jurisdiction.

1. Introdução

Nos últimos anos, a crescente interconectividade de nossos dispositivos e sistemas transformou o monitoramento de redes de dados em uma ferramenta crítica para manter a segurança, a integridade e a eficiência de nossos ambientes digitais. No entanto, à medida que avançamos em direção a essa era digital cada vez mais interligada, surgem questões éticas profundas que não podem ser ignoradas. Este artigo se propõe a explorar e analisar as questões éticas relacionadas ao monitoramento de redes de dados, um campo que se estende desde as questões técnicas e operacionais até as complexidades morais e legais.

O monitoramento de redes de dados, embora fundamental para proteger contra ameaças cibernéticas e garantir o funcionamento adequado das redes, muitas vezes coloca em conflito a necessidade de segurança com a preservação da privacidade e dos direitos individuais. A coleta de informações sensíveis, a retenção de dados e o potencial abuso dessas informações são apenas algumas das preocupações éticas que permeiam esse domínio. À medida que a tecnologia avança e as ameaças cibernéticas se tornam mais sofisticadas, encontrar um equilíbrio entre a busca pela segurança e a garantia de princípios éticos e legais torna-se um desafio crucial e em constante evolução.

Neste contexto, abordaremos o monitoramento de rede usando o software Wireshark em um cenário de ataque DDoS do tipo LOIC (Low Orbit Ion Cannon), que é um ataque de inundação de pacotes de rede. Este tipo de ataque tem o potencial de afetar severamente um usuário desprotegido, ou seja, alguém sem medidas de segurança, como um firewall ou antivírus, configurados em sua máquina. É importante ressaltar que as atividades de monitoramento e análise de tráfego ocorrerão em um ambiente controlado e seguro, utilizando máquinas virtuais para simular o ataque e garantindo que os testes sejam conduzidos de maneira ética e dentro dos limites legais.

2. Referencial Teórico

CARMONA e HEXSEL (2005) definem "Uma rede de computadores é o conjunto de computadores e canais, físicos que compõem um sistema de transporte eletrônico de informações".

2.1. Monitoramento de redes

De acordo com Menezes (2020 p .11 - 12), o referencial teórico aborda o monitoramento de redes de dados como uma parte essencial da segurança cibernética e da gestão de sistemas de informação. Este campo está inserido em um contexto complexo de considerações éticas que se desenvolvem em resposta à evolução tecnológica e à crescente interconectividade global. Privacidade e Direitos Individuais desempenham um papel crucial nesse cenário, uma vez que o monitoramento de redes de dados pode impactar diretamente a liberdade e a privacidade das pessoas. Essas preocupações tornam-se ainda mais prementes com a crescente atenção à regulamentação de proteção de dados, como a LGPD (Lei Geral de Proteção de Dados), que estabelece diretrizes rigorosas para a coleta e o tratamento de informações pessoais.

Além disso, abordaremos as perspectivas éticas do utilitarismo, que permite avaliar as ações de monitoramento com base nas consequências e benefícios gerais, buscando o equilíbrio entre segurança cibernética e preservação da privacidade. A ética Deontológica, que se concentra na moralidade intrínseca das ações, será explorada em relação ao monitoramento de redes de dados, considerando se essas práticas são eticamente aceitáveis, independentemente de suas consequências. A ética da Virtude também desempenha um papel importante, incentivando o desenvolvimento de qualidades virtuosas, como transparência, honestidade e responsabilidade, no contexto do monitoramento de redes de dados.

2.2. DDOS

De acordo com Tecnoblog (2023), um ataque de negação de serviço distribuído, também conhecido como DDoS (Distributed Denial of Service), é uma tática que utiliza múltiplas fontes para sobrecarregar e comprometer um serviço online. Essa técnica explora as limitações de um servidor, inundando-o com solicitações simultâneas de diversas origens. O resultado é a indisponibilidade do serviço, causando prejuízos significativos. Este tipo de ataque envia um fluxo intenso de tráfego de múltiplas origens, sobrecarregando o serviço com solicitações contínuas e criando um tráfego falso, cujo objetivo é exceder a capacidade do site, tornando-o inoperante. Em alguns casos, os atacantes podem exigir um resgate financeiro para interromper o ataque, enquanto em outros, o objetivo é simplesmente prejudicar ou minar a credibilidade do serviço.

Para realizar um ataque DDoS com sucesso, os perpetradores frequentemente contam com "auxiliares", que são recrutados para formar uma botnet. O termo "botnet" é uma combinação das palavras "bot" (robô) e "network" (rede) e descreve uma rede de dispositivos, incluindo computadores, dispositivos móveis e aparelhos conectados à internet, todos infectados por malwares. Esses "escravos virtuais" são controlados pelo atacante, muitas vezes sem o conhecimento dos proprietários, e são usados para conduzir o ataque DDoS.

2.3. HTTP/HTTPS

De acordo com Hostgator Blog (2023), este protocolo desempenha um papel fundamental na transmissão de páginas web na Internet. É importante compreender suas características e especificações para sua utilização eficaz. Funciona em redes baseadas em IP, permitindo a transmissão de páginas web de um servidor para um navegador. Além disso, não se restringe a aplicativos e não emprega criptografia.

O protocolo é sem estado, o que significa que os dados podem ser transmitidos em uma rede IP sem reter informações de estado anterior. Ele é amplamente utilizado para transferir páginas da Internet e dados entre servidores web e navegadores, mas também encontra aplicação em outros contextos.

Por exemplo, o protocolo de transferência de arquivos WebDAV, construído com base nesse protocolo, é utilizado para transferir dados de diretórios e arquivos em redes IP. Além disso, serviços da Web REST têm sua base no Protocolo de Transferência de Hipertexto.

Em relação ao modelo de camadas, o protocolo é associado à camada de aplicação, no sétimo nível. Não emprega criptografia direta, sendo protegido por meio de túneis VPN criptografados (Rede Privada Virtual) ou pela variante criptografada do protocolo HTTPS.

Essa abordagem evita a interceptação de conexões. Quando um site utiliza o protocolo HTTPS, o navegador indica a segurança por meio de um cadeado. Tecnicamente, uma unidade de comunicação em HTTP é denominada mensagem, composta por um cabeçalho e um corpo. O cabeçalho inclui metainformações básicas, como tipo de conteúdo e idioma, enquanto o corpo contém o conteúdo real do site exibido no navegador.

O protocolo TCP/IP é responsável pela transmissão do protocolo, que é sem estado, ou

seja, cada solicitação é independente e processada separadamente, com informações sobre o sucesso da solicitação sendo retornadas.

O protocolo não se limita à transmissão de conteúdo HTML, sendo possível a transmissão de vários formatos de dados. Muitos sites dinâmicos, que utilizam linguagens como PHP, empregam esse protocolo para transmitir formatos diversos.

Dois métodos de solicitação amplamente utilizados são o GET e o POST, com finalidades específicas. O GET busca dados de um servidor, enquanto o POST envia dados para o servidor. Há outros métodos de solicitação, como HEAD, PUT, PATCH, DELETE, CONNECT, OPTIONS e TRACE, cada um com funções distintas (Henrique, 2017).

Motores de busca utilizam o protocolo para recuperar dados de servidores, funcionando com um modelo cliente-servidor, em que o navegador age como cliente, enviando solicitações à porta 80 do servidor, que responde com mensagens.

Alguns sites, como o Google, consideram o protocolo HTTPS como um fator de classificação oficial. Se o arquivo não estiver especificado no endereço, o servidor envia o arquivo padrão do domínio.

O protocolo, embora usado principalmente para transmitir páginas web, não se limita apenas ao hipertexto e é aplicável à troca de diversos tipos de dados. Para garantir transmissões seguras, o protocolo HTTPS é cada vez mais adotado, oferecendo autenticação e criptografia de ponta a ponta, garantindo segurança na Internet. Além disso, o protocolo permite a autenticação do usuário.

2.4. LOIC

Conforme Leoni (2023), este é um software desenvolvido em C# com um único propósito: realizar ataques de negação de serviço (DoS) ou ataques de negação de serviço distribuído (DDoS). Existem também versões em Javascript, chamada JS LOIC, e uma versão web conhecida como Low Orbit Web Cannon. A operação desse software envolve o envio de pacotes TCP e UDP, bem como solicitações HTTP direcionadas a um IP específico e a uma porta predefinida.

Para promover um ataque LOIC, é necessário que vários usuários executem o LOIC em um único endereço IP. Isso ocorre porque um único usuário executando o LOIC normalmente não é capaz de causar uma negação de serviço significativa. No entanto, quando muitos usuários executam o LOIC simultaneamente, o servidor pode sofrer lentidão devido ao tráfego de rede excepcionalmente alto.

Para prevenir ataques LOIC, grandes provedores de serviços oferecem mecanismos de mitigação de DDoS, que operam no nível do provedor de serviços de internet (ISP). Se um usuário hospeda seu próprio servidor na web, ele pode se defender contra-ataques LOIC usando sistemas de detecção e prevenção de intrusões, como o Snort.

Quando ocorre a detecção de uma tentativa de ataque Loic, o usuário pode filtrar os pacotes originados de IPs específicos. Além disso, como uma medida adicional de prevenção, é possível configurar um firewall para limitar o número de solicitações por minuto. Isso ajudará a filtrar o tráfego de ataque, mas não afetará os usuários legítimos.

3. Metodologia

3.1. Tipo de Pesquisa

De acordo com Tech Tudo (2023), para o estudo do monitoramento de ataques LOIC, será realizada uma pesquisa experimental. Isso envolve a criação de um ambiente controlado e seguro, no qual simulações de ataques LOIC serão conduzidas para fins de monitoramento e análise.

3.2. Coleta de Dados

O estudo envolverá a simulação de ataques LOIC em um ambiente controlado. Isso será feito por meio da execução do software LOIC em máquinas virtuais. Essas simulações fornecerão dados de tráfego de rede que serão monitorados e analisados.

3.3. Uso do Wireshark

O Wireshark será utilizado como a ferramenta principal para o monitoramento e análise do tráfego de rede gerado pelas simulações de ataques LOIC. Ele permitirá a captura de pacotes de rede, análise de protocolos, e identificação de padrões de tráfego.

3.4. Amostra

As máquinas virtuais serão utilizadas para simular ataques LOIC. O tamanho da amostra dependerá das condições do experimento e dos recursos disponíveis.

3.5. Usuários de Máquinas Virtuais

A população em estudo consistirá nas máquinas virtuais utilizadas para a simulação de ataques LOIC.

3.6. Ambiente Controlado

O estudo será conduzido em um ambiente controlado, como um laboratório de segurança cibernética. Foram utilizadas duas máquinas virtuais instanciadas no Oracle VM VirtualBox, ambas configuradas em versão Ubuntu 64-bit, 2 unidades de processadores, 25 GB de armazenamento interno e uma única placa de rede em modo NAT, diferindo no aspecto de memória principal, sendo 6144MB para a atacante e 3620MB.

3.7. Análise de Dados e Ética

Os dados coletados serão analisados usando o Wireshark para identificar padrões de tráfego, características dos ataques LOIC e outros aspectos relevantes. Todas as simulações e análises serão conduzidas de maneira ética e dentro dos limites legais. Não haverá impacto em sistemas ou redes reais, e as atividades serão estritamente para fins de pesquisa.

3.8. Considerações Finais

Os resultados do estudo fornecerão informações relevantes sobre como o Wireshark pode ser usado para monitorar ataques LOIC e os desafios éticos associados a esse tipo de monitoramento.

4. Resultados e Discussões

Em concordância com Orestes (2023), em um estudo dividido em três estágios, a análise foi conduzida em ambientes distintos: primeiro, um ambiente controlado em uma rede interna virtual utilizando Radmin; depois, em uma rede interna local; e, finalmente, na comunicação entre redes por meio de uma URL específica (invasaomaquinaartigo.azurewebsites.net). Durante esses testes, várias abordagens foram empregadas, incluindo análise ICMP, envio de solicitações UDP e TCP.

Uma observação importante foi a diferença notável nas respostas dos sistemas durante os ataques. Especificamente, quando utilizamos o método UDP, a comunicação foi interrompida, resultando na perda de todos os pacotes. Isso ocorreu devido ao tamanho relativamente menor dos pacotes UDP em comparação com o protocolo TCP. Isso permitiu uma maior taxa de envio e consequente saturação da largura de banda disponível. Enquanto no ambiente sofrendo interferência via protocolo TCP, a comunicação não foi interrompida. No entanto, as solicitações TCP ocuparam recursos físicos e lógicos da CPU, o que levou a gargalos de hardware e à lentidão das operações.

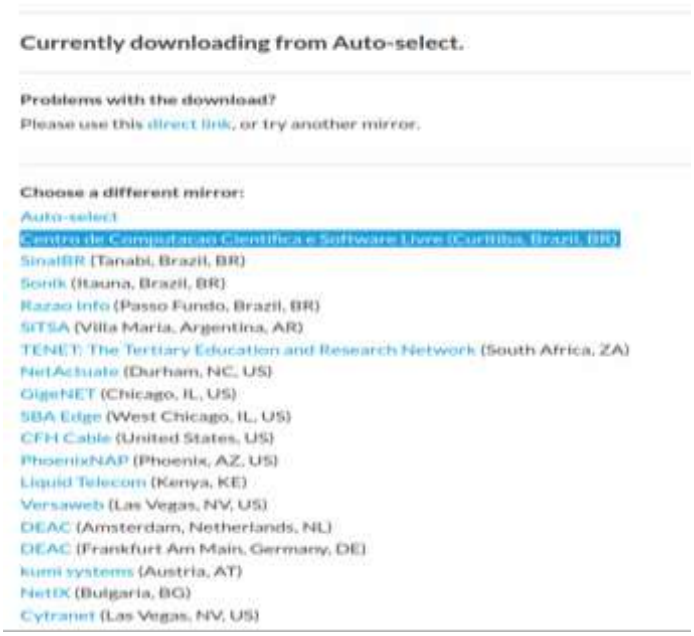
Esses resultados destacam a importância de escolher o protocolo de comunicação apropriado, dependendo dos objetivos e das características do sistema sob teste. A abordagem UDP pode ser mais eficaz em sobrecarregar a rede, interrompendo a comunicação, enquanto o uso de TCP pode não interromper a comunicação, mas pode sobrecarregar o hardware do sistema, prejudicando o desempenho. Essas conclusões podem ser valiosas para a otimização e a segurança de sistemas de rede em diferentes cenários.

Os quocientes foram semelhantes, somente no domínio oferecido pela plataforma Microsoft Azure houve picos, por contar com uma segurança pré-estabelecida interrompendo a inundação de pacotes evidenciado pela sequência repetitiva do endereço ip do invasor, identificando o atacante na rede e sobrepondo as requisições de pacote.

4.1. Preparação do ambiente atacante e Ferramenta LOIC

Os testes e simulações para com a rede interna virtual, vieram a ser realizados em uma aplicação, via *VirtualBox* e um arquivo de imagem *Mint 212 Cinnamon* em 64 bits, exclusivamente para a máquina qual realizaria os ataques; A instalação da ferramenta LOIC foi realizada por meio da *SourceForge*, repositório de código fonte baseado em *Web*, por meio de um espelho do Centro de Computação científica e Sofware Livre, Curitiba, PR, apresentado na Figura 1:

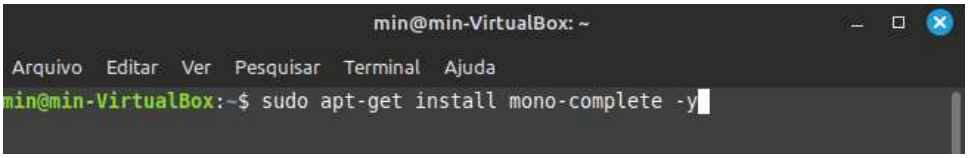
Figura 1 – Download da ferramenta LOIC



Fonte: Autores

Observam-se outras disposições de espelho de Download da ferramenta, sendo optada a versão Brasileira hospedada em Curitiba por conta da melhor conectividade com o canal. Após o download, foram realizadas as atualizações e upgrades no sistema via terminal. A última etapa ante finalização do ambiente, foi a instalação e execução do Mono, plataforma de desenvolvimento e execução de programação C# (C Sharp) com a finalidade de lidar com o LOIC, o qual foi uma implementação originalmente de um código aberto de estrutura .NET da Microsoft. A linha de comando necessária para realização das ações é evidenciada nas Figuras 2, 3, 4 e 5:

Figura 2 - Instalação Mono



Fonte: Autores

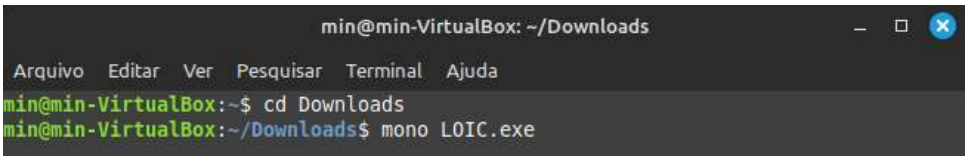
Figura 3 – Update do ambiente

Fonte: Autores

Figura 4 – Upgrade do Sistema

Fonte: Autores

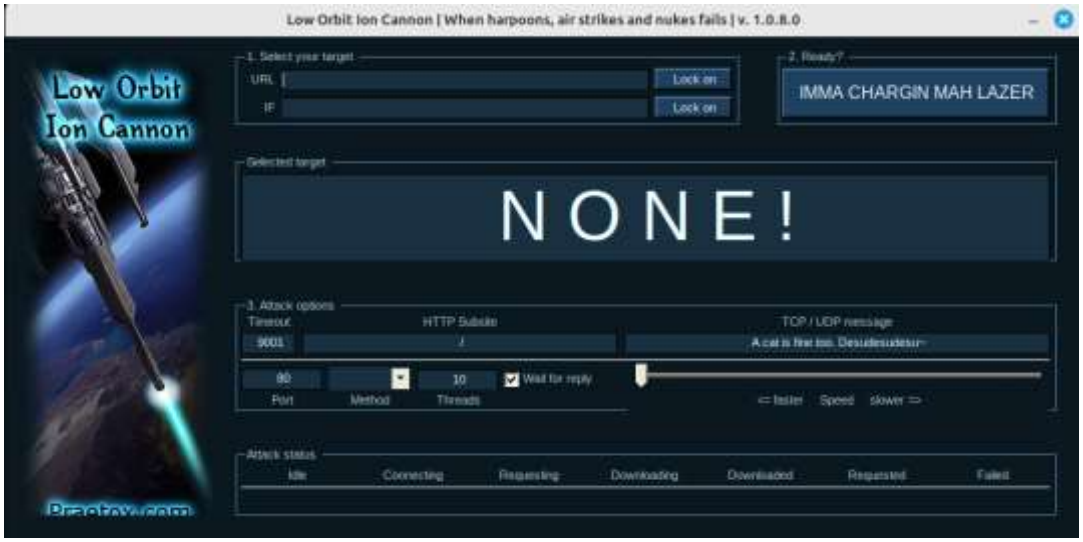
Figura 5 - Execuação da ferramenta LOIC



Fonte: Autores

Os campos disponíveis em 3 principais seções: Select your target, o qual você sinaliza o alvo de ataque, podendo ser por meio de uma URL, IP ou domínio; Ready, ativa o LOIC e inicia o ataque; e Attack options, no qual define o timeout, porta a ser utilizada, método, podendo ser TCP, UDP e HTTP, threads e mensagens a ser enviada junto ao ataque. A interface padrão do LOIC é apresentada na Figura 6:

Figura 6 – Interface do LOIC



Fonte: Autores

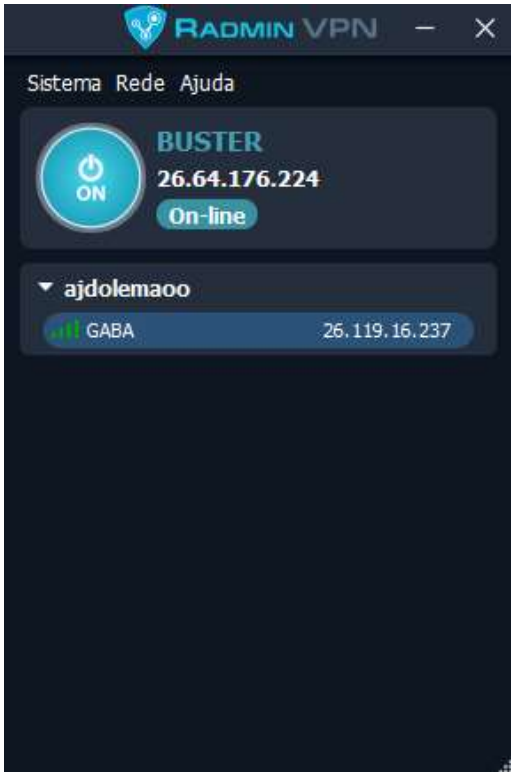
4.2. Estágio em Rede interna virtual

Para realização do estágio 1, a máquina atacante esteve em ambiente compartilhado com o host por meio do Radmin, software de VPN e acesso remoto, disponível de forma gratuita no próprio site da aplicação (versão 1.4.4642.1).

O host de nome “BUSTER” e IP 26.64.176.224 é identificado como o atacante da simulação.

O host de nome “GABA” e IP 26.119.16.237 é a máquina a qual sofrerá o ataque. A configuração do Radmin VPN é apresentada na Figura 7:

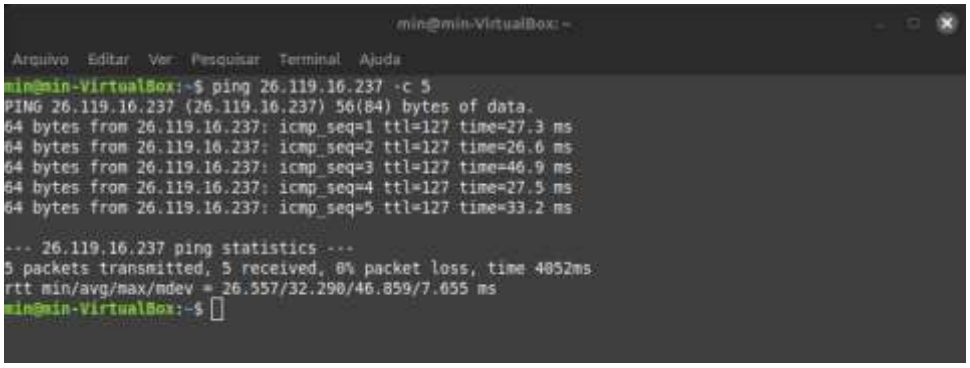
Figura 7 - Configuração RadminVPN



Fonte: Autores

Em um primeiro momento foi realizada uma verificação por PING ao IP, para verificação de conectividade, apesentados na Figura 8. O mesmo método foi utilizado para constatar a conexão durante os ataques, observado nas Figuras 9 e 10;

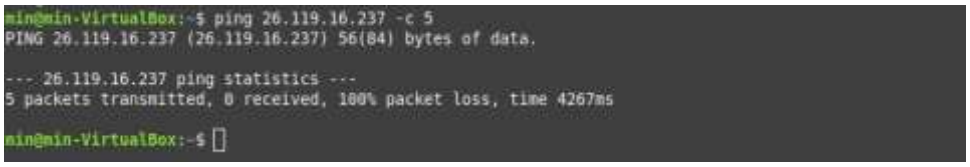
Figura 8 – Protocolo ICMP



Fonte: Autores

Os valores de resultado apresentam um sucesso na comunicação, informando que os 5 sinais enviados foram entregues com tempos de 27.3, 26.6, 46.9 , 27.5 e 33.2 milissegundos. É confirmada o envio dos pacotes nas estáticas apresentadas no final da captura de tela, com 5 envios realizados com sucesso, 0% de perda de pacotes e um tempo total de conversa de 4052 milissegundos. Os mesmos campos vão ser observados em testes futuros.

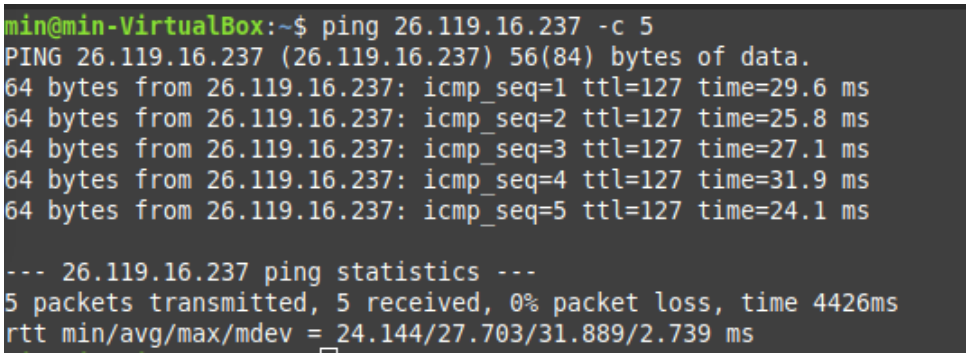
Figura 9 - Verificação ICMP em ataque UDP



Fonte: Autores

A ausência de mensagem de confirmação informa a falha na comunicação com o endereço final., evidenciado nas estáticas do final da captura de tela, sendo, 5 pacotes transmitidos, 0 recebidos pelo endereço totalizando 100% de perda de pacotes e um tempo total de 4267ms da operação. Os testes e verificação de falha na comunicação serão verificados da mesma forma.

Figura 10 - Verificação ICMP em ataque TCP



Fonte: Autores

Os testes de ataque realizados foram baseados em utilização da porta 80 e métodos TCP e UDP, focos de análise; Resultados expressos nas Figuras 11 e 12:

Figura 11 – LOIC em ataque UDP

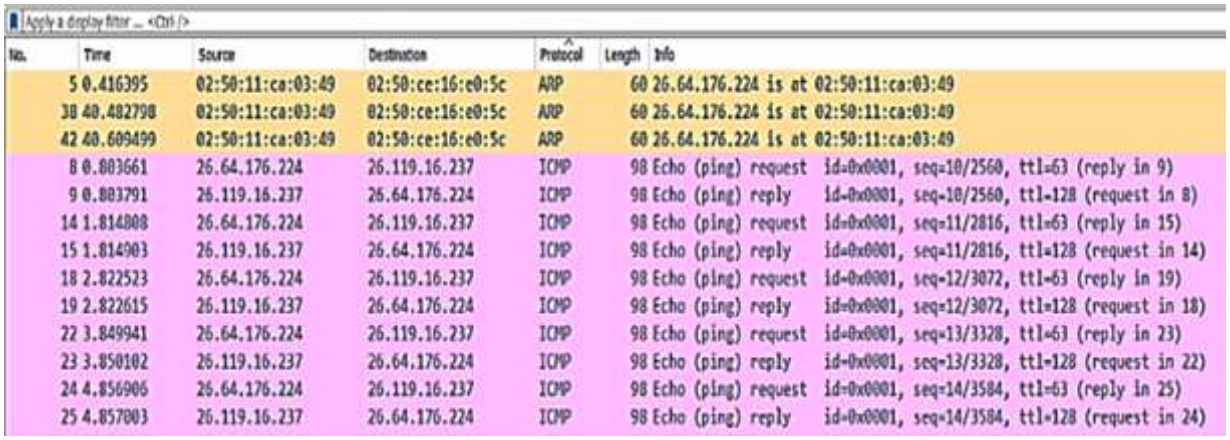
Fonte: Autores

Figura 12 – LOIC em ataque TCP

Fonte: Autores

O monitoramento e visualização dos ataques, foram realizadas por uma terceira máquina, conectada ao host a ser monitorado. O software de gerenciamento utilizado foi o Wireshark. Os resultados dos métodos ICMP assim como os do ataque, são visualizados nas Figuras 13, 14 e 15:

Figura 13 – Monitoramento ao protocolo ICMP

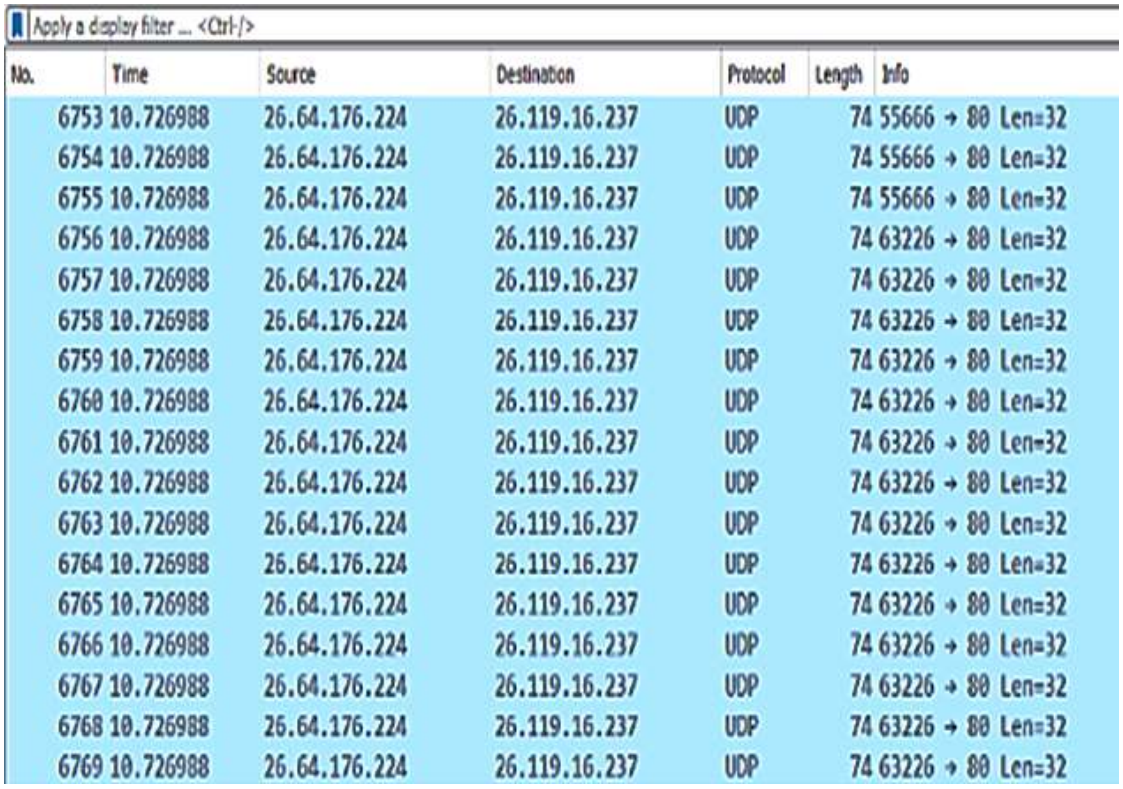


No.	Time	Source	Destination	Protocol	Length	Info
5	0.416395	02:50:11:ca:03:49	02:50:ce:16:e0:5c	ARP	60	26.64.176.224 is at 02:50:11:ca:03:49
38	40.482798	02:50:11:ca:03:49	02:50:ce:16:e0:5c	ARP	60	26.64.176.224 is at 02:50:11:ca:03:49
42	40.609499	02:50:11:ca:03:49	02:50:ce:16:e0:5c	ARP	60	26.64.176.224 is at 02:50:11:ca:03:49
8	0.803661	26.64.176.224	26.119.16.237	ICMP	98	Echo (ping) request id=0x0001, seq=10/2560, ttl=63 (reply in 9)
9	0.803791	26.119.16.237	26.64.176.224	ICMP	98	Echo (ping) reply id=0x0001, seq=10/2560, ttl=128 (request in 8)
14	1.814808	26.64.176.224	26.119.16.237	ICMP	98	Echo (ping) request id=0x0001, seq=11/2816, ttl=63 (reply in 15)
15	1.814903	26.119.16.237	26.64.176.224	ICMP	98	Echo (ping) reply id=0x0001, seq=11/2816, ttl=128 (request in 14)
18	2.822523	26.64.176.224	26.119.16.237	ICMP	98	Echo (ping) request id=0x0001, seq=12/3072, ttl=63 (reply in 19)
19	2.822615	26.119.16.237	26.64.176.224	ICMP	98	Echo (ping) reply id=0x0001, seq=12/3072, ttl=128 (request in 18)
22	3.849941	26.64.176.224	26.119.16.237	ICMP	98	Echo (ping) request id=0x0001, seq=13/3328, ttl=63 (reply in 23)
23	3.850102	26.119.16.237	26.64.176.224	ICMP	98	Echo (ping) reply id=0x0001, seq=13/3328, ttl=128 (request in 22)
24	4.856906	26.64.176.224	26.119.16.237	ICMP	98	Echo (ping) request id=0x0001, seq=14/3584, ttl=63 (reply in 25)
25	4.857003	26.119.16.237	26.64.176.224	ICMP	98	Echo (ping) reply id=0x0001, seq=14/3584, ttl=128 (request in 24)

Fonte: Autores

Os campos em lilás sugerem os testes PING realizados pela máquina atacante, verificando e confirmando comunicação. Através da coluna "*source*" podemos verificar o ponto do qual os pacotes são enviados.

Figura 14 – Monitoramento UDP



Fonte: Autores

Os campos em azul demonstram o recebimento de pacotes no formato UDP, como resultado da inundação do protocolo, pelo software LOIC.

Figura 15 – Monitoramento TCP

No.	Time	Destination	Protocol	Length	Info
33	7.277688	26.64.176.224	TCP	54	80 → 52518 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
38	7.751224	26.119.16.237	TCP	66	52520 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
39	7.751257	26.64.176.224	TCP	54	80 → 52520 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
40	7.778021	26.119.16.237	TCP	66	[TCP Retransmission] 52519 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
41	7.778577	26.64.176.224	TCP	54	80 → 52519 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
42	8.252594	26.119.16.237	TCP	66	52521 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
43	8.252649	26.64.176.224	TCP	54	80 → 52521 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
44	8.277485	26.119.16.237	TCP	66	[TCP Retransmission] 52520 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
45	8.277527	26.64.176.224	TCP	54	80 → 52520 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
50	8.755369	26.119.16.237	TCP	66	52522 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
51	8.755415	26.64.176.224	TCP	54	80 → 52522 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
52	8.777601	26.119.16.237	TCP	66	[TCP Retransmission] 52521 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
53	8.777645	26.64.176.224	TCP	54	80 → 52521 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
54	9.254389	26.119.16.237	TCP	66	52523 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
55	9.254416	26.64.176.224	TCP	54	80 → 52523 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
56	9.280317	26.119.16.237	TCP	66	[TCP Retransmission] 52522 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
57	9.280371	26.64.176.224	TCP	54	80 → 52522 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
60	9.782172	26.119.16.237	TCP	66	[TCP Retransmission] 52523 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
61	9.782208	26.64.176.224	TCP	54	80 → 52523 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
64	13.247391	26.119.16.237	TCP	66	52515 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
65	13.247421	26.64.176.224	TCP	54	80 → 52515 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
66	13.247473	26.119.16.237	TCP	66	52514 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
67	13.247479	26.64.176.224	TCP	54	80 → 52514 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

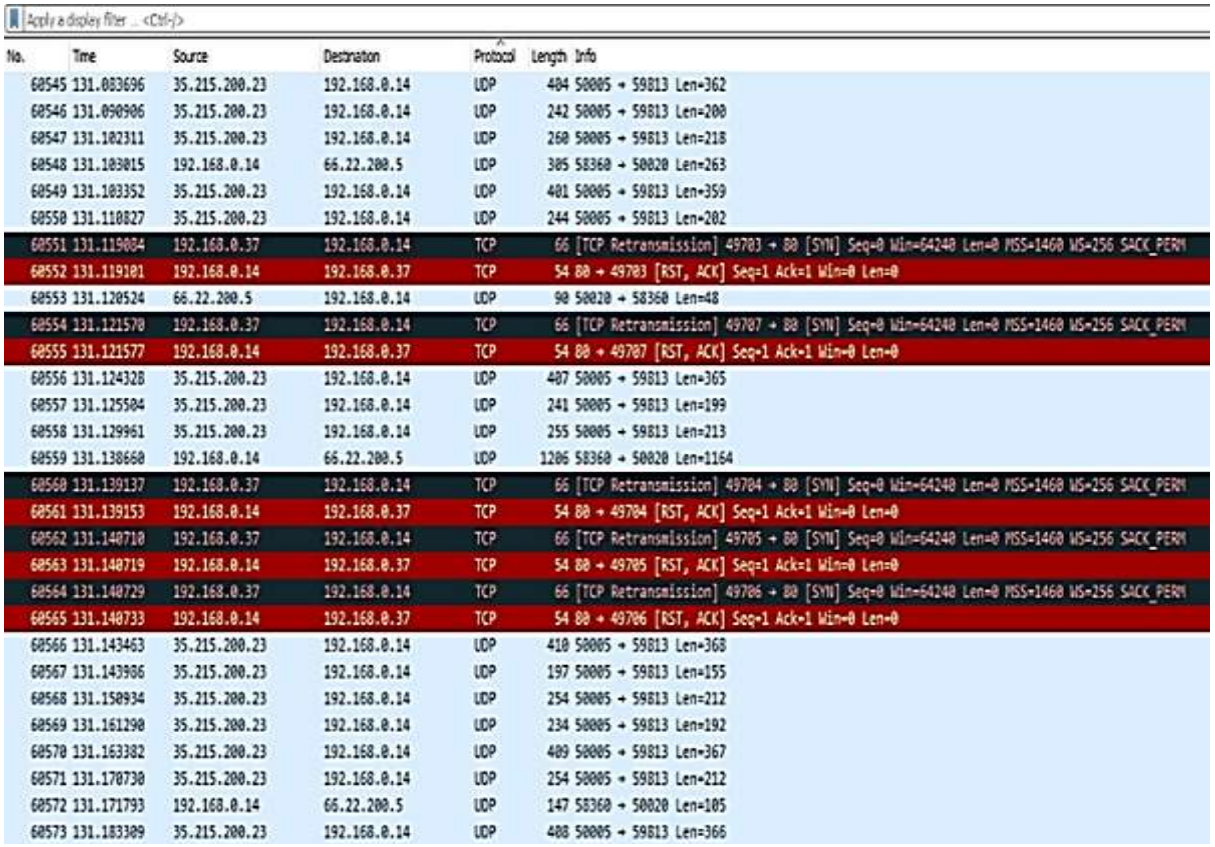
Fonte: Autores

As entradas em amarelo, preto e vermelho representam o protocolo TCP.

4.3. Estágio em Rede interna

Em um segundo momento, os mesmos testes de ataque TCP e UDP foram realizados em ambiente local, utilizando da mesma ferramenta para ataque (LOIC) e monitoramento (Wireshark). Os resultados observados foram semelhantes, no sentido da consequência à máquina sob ataque. Em função do ambiente local, não foi necessária a utilização do Radmin. O host de IP 192.168.0.37 é identificado como host atacante e a de IP 192.168.0.14 como atacada. Resultado representado na Figura 16:

Figura 16 - Monitoramento TCP e UDP



Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
60545	131.083696	35.215.200.23	192.168.0.14	UDP	404	50005 → 59813 Len=362
60546	131.090906	35.215.200.23	192.168.0.14	UDP	242	50005 → 59813 Len=200
60547	131.102311	35.215.200.23	192.168.0.14	UDP	260	50005 → 59813 Len=218
60548	131.103015	192.168.0.14	66.22.200.5	UDP	305	58360 → 50020 Len=263
60549	131.103352	35.215.200.23	192.168.0.14	UDP	401	50005 → 59813 Len=359
60550	131.110827	35.215.200.23	192.168.0.14	UDP	244	50005 → 59813 Len=202
60551	131.119084	192.168.0.37	192.168.0.14	TCP	66	[TCP Retransmission] 49703 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
60552	131.119101	192.168.0.14	192.168.0.37	TCP	54	80 → 49703 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
60553	131.120524	66.22.200.5	192.168.0.14	UDP	90	50020 → 58360 Len=48
60554	131.121570	192.168.0.37	192.168.0.14	TCP	66	[TCP Retransmission] 49707 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
60555	131.121577	192.168.0.14	192.168.0.37	TCP	54	80 → 49707 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
60556	131.124328	35.215.200.23	192.168.0.14	UDP	407	50005 → 59813 Len=365
60557	131.125504	35.215.200.23	192.168.0.14	UDP	241	50005 → 59813 Len=199
60558	131.129961	35.215.200.23	192.168.0.14	UDP	255	50005 → 59813 Len=213
60559	131.138660	192.168.0.14	66.22.200.5	UDP	1206	58360 → 50020 Len=1164
60560	131.139137	192.168.0.37	192.168.0.14	TCP	66	[TCP Retransmission] 49704 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
60561	131.139153	192.168.0.14	192.168.0.37	TCP	54	80 → 49704 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
60562	131.140710	192.168.0.37	192.168.0.14	TCP	66	[TCP Retransmission] 49705 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
60563	131.140719	192.168.0.14	192.168.0.37	TCP	54	80 → 49705 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
60564	131.140729	192.168.0.37	192.168.0.14	TCP	66	[TCP Retransmission] 49706 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
60565	131.140733	192.168.0.14	192.168.0.37	TCP	54	80 → 49706 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
60566	131.143463	35.215.200.23	192.168.0.14	UDP	410	50005 → 59813 Len=368
60567	131.143986	35.215.200.23	192.168.0.14	UDP	197	50005 → 59813 Len=155
60568	131.150934	35.215.200.23	192.168.0.14	UDP	254	50005 → 59813 Len=212
60569	131.161290	35.215.200.23	192.168.0.14	UDP	234	50005 → 59813 Len=192
60570	131.163382	35.215.200.23	192.168.0.14	UDP	409	50005 → 59813 Len=367
60571	131.170730	35.215.200.23	192.168.0.14	UDP	254	50005 → 59813 Len=212
60572	131.171793	192.168.0.14	66.22.200.5	UDP	147	58360 → 50020 Len=105
60573	131.183309	35.215.200.23	192.168.0.14	UDP	408	50005 → 59813 Len=366

Fonte: Autores

4.4. Estágio web

A última bateria de testes foi realizada utilizando dessa vez, o serviço de rede e gerenciamento web da Azure; Fora utilizado o recurso gratuito da nuvem, para publicação e hospedagem de um domínio denominado “invasaomaquinaartigo.azurewebsites.net” e IP 20.206.176.5 O monitoramento dessa vez foi de forma visual através dos gráficos oferecidos pelo serviço. O protocolo ICMP foi utilizado em simultâneo à supervisão gráfica, para verificação simplificada do serviço. Os resultados do monitoramento são apresentados nas Figuras 17 e 18:

Figura 17 – Monitoramento Azure entrada/saída - UDP e TCP

Fonte: Autores

Figura 18 – Visualização do ICMP durante testes

```
min@min-VirtualBox:~$ ping 20.206.176.5
PING 20.206.176.5 (20.206.176.5) 56(84) bytes of data.
64 bytes from 20.206.176.5: icmp_seq=1 ttl=113 time=5.28 ms
64 bytes from 20.206.176.5: icmp_seq=2 ttl=113 time=5.78 ms
64 bytes from 20.206.176.5: icmp_seq=3 ttl=113 time=5.86 ms
64 bytes from 20.206.176.5: icmp_seq=4 ttl=113 time=5.41 ms
64 bytes from 20.206.176.5: icmp_seq=5 ttl=113 time=5.06 ms
64 bytes from 20.206.176.5: icmp_seq=6 ttl=113 time=4.79 ms
64 bytes from 20.206.176.5: icmp_seq=7 ttl=113 time=5.24 ms
64 bytes from 20.206.176.5: icmp_seq=8 ttl=113 time=25.1 ms
64 bytes from 20.206.176.5: icmp_seq=9 ttl=113 time=69.6 ms
64 bytes from 20.206.176.5: icmp_seq=10 ttl=113 time=68.5 ms
64 bytes from 20.206.176.5: icmp_seq=11 ttl=113 time=105 ms
64 bytes from 20.206.176.5: icmp_seq=12 ttl=113 time=130 ms
64 bytes from 20.206.176.5: icmp_seq=22 ttl=113 time=72.1 ms
64 bytes from 20.206.176.5: icmp_seq=23 ttl=113 time=5.86 ms
64 bytes from 20.206.176.5: icmp_seq=24 ttl=113 time=4.98 ms
```

Fonte: Autores

5. Considerações Finais

Este artigo traz uma visão de mapear os danos advindos de ataques cibernéticos dentro de um ambiente de rede em tempo real, usando softwares apresentados em aula. Ao término deste trabalho evidencia-se a enorme importância de se atuar com exatidão e técnicas apuradas no monitoramento de redes.

Assim como está descrita uma forma de se agir com uma conduta pautada na legislação vigente. Deste modo, essas ferramentas de gerência de rede tem a finalidade de orientar e trazer parâmetros para um maior controle e uma melhor tomada de decisão sobre as adversidades apresentadas.

Com a demonstração de dispositivos de segurança, o trabalho atinge seus objetivos ao evidenciar que um bom planejamento já que este será o caminho do alcance do objetivo final. Sabendo-se como coletar os dados, como discriminá-los, analisá-los com critério sempre pode trazer benefícios sem precedentes.

Cumpre-se a proposta de mostrar como se administrar problemas de ataques contra uma rede consolidada de dados, através de ações tomadas pela por parte da gerência de redes de dados da empresa e deste modo, proporcionar Maior segurança para as operações e informações corporativas. Nota-se que, a ciência da computação desde o os primórdios da era moderna, atua para a fixação de se ser um ramo de atuação confiável e autêntico.

Referências

ORESTES, Yan. O que é UDP e TCP? Entenda quais as diferenças e como funciona cada Protocolo. Alura, 18, set, 2023. Disponível em: https://www.alura.com.br/artigos/quais-as-diferencas-entre-o-tcp-e-o-udp

MENEZES, Rafael. MONITORAMENTO VOLTADO À CIBERSEGURANÇA EM SISTEMAS INDUSTRIAIS. Universidade tecnologia federal do Paraná, 18, nov, 2020. p. 11 - 12.

TECNOBLOG. O que é um ataque DDoS? TECNOBLOG, 2023 Disponível em: https://tecnoblog.net/?s=Http, Acesso em 18, set, 2023.

Leoni, Joana. DIO. Ataque Loic: o que é, como funciona e como se prevenir DIO, 2023 Disponível em: https://www.dio.me/articles/ataque-loic-o-que-e-como-funciona-e-como-se-prevenir, Acesso em 21, set, 2023.

HOSTGATOR BLOG. Protocolo HTTP: entenda o que é e para que serve! HOSTGATOR BLOG, 2023 Disponível em: https://www.hostgator.com.br/blog/o-que-e-protocolo-http/, Acesso em 18, set, 2023.

TECH TUDO. Como usar o Wireshark TECH TUDO, 2023 Disponível em: https://www.techtudo.com.br/noticias/2012/09/como-usar-o-wireshark.ghtml,

Acesso em 23, set, 2023.

Henrique. ASP.NET MVC: como utilizar os métodos HTTP, 2017 Disponível em: https://www.devmedia.com.br/asp-net-mvc-como-utilizar-os-metodos-http/37893#:~:text=GET%3A%20M%C3%A9todo%20gen%C3%A9rico%20para%20qualquer,remo%C3%A7%C3%A3o%20de%20dados%20no%20servidor , Acesso em 24, set, 2023.