Testes de Invasão: Ciclo de Vida e Laboratório Seguro

Laila Ohanna Orsay Ferreira, Sandro R. Fernandes

Instituto Federal do Sudeste de Minas Gerais

Campus Juiz de Fora – MG – Brasil

laila.orsayferreira@gmail.com, sandro.fernandes@ifsudestemg.edu.br

Abstract. *The widespread use of information systems brings forth a wide array of security flaws. Lack of defense systems, programs that have error codes or that were created without security policies leave a system vulnerable to numerous types of threats that may compromise its use. As a way of broadening the understandment on denial of services (DoS), which pose great risk to information systems, this paper proposes to analyze this threat, utilizing a virtual enviroment and methodology for pentests, in a practical and ethical manner.*

Resumo. *A ampla utilização de sistemas de informação traz à tona uma gama de brechas em sua segurança. Falta de defesa, utilização de programas que possuem códigos com erros ou que foram criados sem políticas de segurança deixam um sistema vulnerável a inúmeros tipos de ameaças que podem comprometer a utilização de seus serviços. A fim de criar maior entendimento sobre ataques de negação de serviço, que definem maiores riscos a sistemas, foi proposto um laboratório virtual e metodologia para pentests, de forma prática e ética.*

1 Introdução

A arquitetura da Internet expõe seus usuários a uma série de vulnerabilidades que afetam o desempenho de máquinas e servidores, o que pode ocasionar uma série de danos. Negligenciando estas vulnerabilidades podem ocorrer falhas na disponibilidade de algum serviço que podem ter desde pequenas repercussões aos serviços prestados quanto, por exemplo, no caso de ser do ramo financeiro, resultar em grandes perdas monetárias [Maciel, 2018]. Uma das

formas em que o usuário fica exposto é sob um ataque de DoS, *Denial of Service* - Negação de Serviço, que tem se tornado cada vez mais comum. O tempo de indisponibilidade de alguns serviços causados por esse tipo de ataque chega a ⅓ do total [Bardal 2015]. Tais ataques consomem recursos dos servidores e roteadores e impedem que usuários legítimos tenham acesso a um determinado serviço. Prevenir esse tipo de ataque é um desafio mesmo para profissionais de TI que mantêm seus computadores e/ou servidores regularmente atualizados e bem supervisionados. Realizando testes de invasão em servidores reais, são descobertos problemas de segurança que podem ser corrigidos.

1.1 Objetivo

Este trabalho propõe uma análise da metodologia de um ataque de negação de serviço, assim como a proposta de um ambiente seguro para testar esse tipo de ataque, realizada através de um laboratório virtual que simule um ataque mais próximo à realidade.

2 Revisão Sistemática

A prática baseada em evidências é o uso das melhores evidências científicas para apoiar a adoção de decisões. Identificar a evidência requer adequada construção da pergunta de pesquisa e de revisão da literatura foi utilizado a estratégia PICO para a construção da pergunta de pesquisa e busca bibliográfica [Santos *et al*. 2007]. A Tabela 1 abaixo descreve o PICOC construído para este trabalho:

PICOC	**Palavras-chave**
População	DDoS, DoS, ataque de negação de serviço
Intervenção	comparação, estudo de caso, testes e simulação
Comparação	ataques Smurf e Syn Flood
Resultado	estudo de caso, análise

Contexto	Acadêmicos e Profissionais de TI

Tabela 1. PICOC

A partir do escopo deste trabalho, define-se a questão de pesquisa: “Existe hoje uma base de dados atual que represente bem o contexto de um teste de invasão e estrutura de forma simples e segura que suporte diferentes tipos de testes para ataque?”.

2.1 Protocolo para Revisão Sistemática

A partir da definição da questão de pesquisa, foram utilizadas principalmente duas *strings* de busca:

a) (DDoS OR *denial of service)* AND *attack* AND *pentest* AND *lab* AND *virtualization*

b) (DDoS OR *denial of service)* AND *case study* OR *case study research* AND *lab structure* AND *virtual machine*

Além disso, foi feita uma estratégia de busca visando a escolha das pesquisas mais relevantes sobre o tema. Os estudos que foram escolhidos passaram pelos critérios de inclusão e exclusão [Donato, 2019] da tabela abaixo, após leitura e verificação de sua relevância:

Questão de pesquisa	Existe hoje uma base de dados atual que represente bem o contexto de um teste de invasão e estrutura de forma simples e segura que suporte diferentes tipos de testes para ataque?

Métodos de busca de fontes	Com o uso de palavras-chaves definidas, buscas por artigos de periódicos ou trabalhos de eventos científicos ou da área de graduação, mestrado ou pós-graduação serão ser realizadas
Palavras-chave definidas	Denial of service, DoS, DDoS, study case, virtualization
Listagem de fontes	IEEE, Google Acadêmico
Idiomas	Português e inglês
Critérios de inclusão	Estudos sobre ataques de negação de serviço, trabalhos na área de segurança da informação que abordem o tema de ataque e/ou estudos de casos com laboratórios virtualizados
Critérios de exclusão	Serão desconsiderados artigos que não estejam integralmente nas bases de dados escolhidas, que possuam experimentos práticos, sem desenvolvimento ou não aborda funções de similaridade
Tipo de artigos	Artigos da área de segurança da informação que contenham experimentos reais, estatísticas, que possuam similaridade com o tema do presente estudo

Tabela 2. Protocolo Revisão Sistemática

Como resultado de busca nas bases de dados Google Acadêmico e Periódico Capes foram encontrados 8.354 artigos científicos, dos quais, após passarem pelos critérios descritos acima, 4 artigos foram selecionados. Feito o levantamento dos estudos encontrados, para filtrar os artigos que possuíam maior grau de relevância para a pesquisa, foi obedecida a presença dos seguintes critérios:

1. Descrição do ambiente utilizado durante os experimentos
2. Testes de invasão em ambiente virtualizado
3. Descrição de tipos de ataque realizado
4. Descrição de estrutura de um *pentest*

Mesmo com os artigos selecionados foram utilizados 3 (três) livros que complementam a fundamentação teórica e que foram adicionados nas referências bibliográficas.

2.2 Trabalhos relacionados

Rafael Machado Bardal (2014) apresenta de forma mais aprofundada sobre o histórico do avanço desse tipo de ameaça ao longo dos anos, além de apresentar um laboratório virtualizado para realizar ataques *Smurf* e inundação SYN, mostrando que ambientes virtualizados são eficazes para testes de invasão.

Francisco R. C. Araújo et al (2014) apresenta em seu artigo os benefícios da utilização de um ambiente virtual para realizar testes de Infraestrutura de Redes.

Rafael E. C. Santos (2018) fala sobre as grandes falhas existentes na segurança, além de descrever as fases de um ataque de negação de serviço e mostra que o *pentest* é uma prática necessária para que instituições, agências e empresas descubram vulnerabilidades para entender
erros e falhas existentes e buscar soluções para contornar ou mitigar problemas dentro da lei.

Monteverde (2014), realizou um estudo de vulnerabilidade de serviços web em

uma amostra de sites brasileiros usando scanners de vulnerabilidade automatizados. Como pode ser visto neste trabalho, 33% das vulnerabilidades encontradas nos testes são classificadas como críticas e ainda podem ser facilmente exploradas no trabalho por alguém com algum conhecimento técnico e usando uma ferramenta que fornece uma interface gráfica.

3 Fundamentação Teórica

3.1 Denial of Service - Negação de Serviço

Ataques DoS, também conhecidos como ataques de Negação de Serviço, são ataques a computadores ou servidores WEB projetados para impedi-los de serem usados pelos usuários [Weidman, 2015].

Dessa forma, exploraram vulnerabilidades em certos sistemas de servidor, e, portanto, quando diversas solicitações são feitas, os recursos do servidor, tais como memória ou largura de banda, foram esgotados, de modo que os serviços fornecidos pelo servidor não estejam mais em funcionamento [Zuben, 2016].

Outro tipo de ataque DoS são os ataques de consumo aos recursos de hardware. Esse tipo de ataque tenta consumir alguns recursos, como CPU e memória ou de equipamentos de rede (roteadores e *Firewalls*); para este caso podemos citar como exemplo *SYN Flood* [Zuben, 2016].

Por último pode-se citar os ataques volumétricos, que possuem como objetivo consumir os recursos de largura de banda de um link, utilizando botnets, máquinas comprometidas e utilizam equipamentos mal configurados que permitem a amplificação de requisições, fazendo o *Spoof* (imitação) do IP da vítima para forçar respostas amplificadas a ela [Zuben, 2016].

3.2 Ciclo de vida de um teste de invasão

A fim de garantir uma abordagem mais fundamentada, foi utilizado um *framework* desenvolvido a fim de que hackers éticos pudessem se pautar para realizar seus testes. Os quatro passos que compõem o processo são: Reconhecimento, *Scanning*, Exploração de Falhas e Preservação do Acesso [Engebretson, 2013].

3.2.1 Reconhecimento

Nesta primeira fase o hacker ético adquire o máximo de informações sobre o alvo no geral (empresa e sistema), que facilitará a exploração de vulnerabilidades mais tarde. Esse reconhecimento pode ser feito de diversas formas, incluindo pesquisas online utilizando a busca avançada do Google e posts de redes sociais, o que facilita a utilização de uma engenharia social, por exemplo. Algumas informações que estão no próprio site da empresa são essenciais, por exemplo, em uma página de carreiras, disponibilizar as tecnologias que são utilizadas no desenvolvimento ou até mesmo seus organogramas e perfil de líderes internos. [Broad, Bindner; 2015]

3.2.2 Scanning

Seguinte ao reconhecimento, é feito o *scanning*, onde o hacker utiliza as informações adquiridas na fase de reconhecimento para saber mais sobre os dispositivos conectados à rede alvo. Determina-se o tipo de aparelho (notebook, desktop, móvel), sistema operacional, aplicações web e possíveis vulnerabilidades [Broad, Bindner; 2015]. Essa segunda fase tem como objetivo obter os possíveis alvos para a exploração de falhas. Algumas ferramentas podem ser utilizadas para auxiliar no scanning, como, por exemplo, a Nmap, que determina as máquinas ativas e seus respectivos sistemas operacionais, assim como portas que estão ouvindo e, talvez, credenciais dos usuários, sendo uma importante ferramenta nessa fase [Monteverde, 2014].

3.2.3 Exploração de falhas (*exploitation*)

Como definido pelo NIST (*National Institute of Science and Technology*), Publicação Especial 800-30, Apêndice B, página B-133, uma vulnerabilidade é definida como “um ponto fraco em um sistema de informação, nos procedimentos de segurança de um sistema, nos controles internos ou em uma implementação, e que pode ser explorado por uma fonte de ameaças" [Broad, Bindner; 2015]. De uma forma mais literal, a vulnerabilidade é causada por uma falha que pode existir em qualquer parte do sistema, máquina e/ou ambiente tecnológico, e pode decorrer de ações de terceiros que administram o sistema de informação. A exploração de falhas é identificar esses possíveis erros e assim atacá-los de forma

mais eficaz. Como a exploração de falhas tende a ser algo quase ilimitado — tendo em vista que existem diversas portas de entrada —, utilizar uma ferramenta como a *Metasploitable* pode ajudar com a pesquisa de exploits locais e remotos (quando um exploit tem como alvo um dispositivo a partir de local externo ao sistema operacional base, uma *Smart TV* por exemplo) [Engebretson, 2013].

3.2.4 Preservação do acesso

A fase pós exploratória é considerada a mais importante de um teste de invasão, cujo maior objetivo é preservar o acesso ao sistema vulnerável. De forma resumida, a preservação do acesso pode ser feita de diversas formas, mas consiste basicamente em reduzir os esforços para atacar o ponto considerado vulnerável. Nesta fase é possível explorar os diversos dados sensíveis do sistema invadido: arquivos confidenciais, acesso a rede ou um domínio que dê acesso a outros sistemas do mesmo domínio. [Weidman, 2015]. Uma forma de preservar o acesso são *malwares*, que é um tipo de código malicioso que engloba vírus, trojans e vírus. Um Cavalo de Troia (*Trojan*), por exemplo, pode ser usado a fim de preservar o acesso executando a ação desejada, porém de forma oculta com finalidade de criar *backdoors*— programas que são deixados em execução a fim de facilitar possíveis entradas posteriores ao sistema, excluindo a necessidade de uma exploração de falhas prévia —, executar scripts e roubar dados.

Outra forma bastante comum são os vírus, que, diferente do *Trojan*, são códigos maliciosos que infectam processos ou arquivos que podem se estender a arquivos e algumas partes do hardware [Weidman, 2015].

4 Metodologia

A seguir serão propostas algumas simulações de ataque em laboratório, pois são práticas a serem realizadas em ambiente controlado, a fim de não causar danos a serviços essenciais ou ambientes de produção.

A virtualização possui diversas vantagens, que podem ser destacadas: segurança, disponibilidade, custo, adaptabilidade a diferentes cargas de trabalho, balanceamento de carga e suporte para aplicativos legados [Menascé, 2005].

O Brasil possui diversas leis regulamentadoras da área da computação, como, dentro do Código Penal Brasileiro, como por exemplo o artigo 154-A que aborda a invasão de computadores, clonagem de dados, *botnets* zumbis, propagação de vírus ou o 313-B que aborda a modificação não autorizada de sistema [Emerick, 2018]. Dessa forma, a virtualização torna-se indispensável para fins didáticos.

Para a criação do laboratório, iremos utilizar um ambiente virtualizado no Oracle VM Virtualbox em um computador pessoal, possuindo como processador Intel(R) Core (TM) i5-5200U CPU funcionando a 2.20 GHz, com 8GB de RAM e placa de vídeo NVIDIA GeForce 920m, sistema operacional Windows 10 Home 64 bits, SSD SanDisk 120GB. O roteador utilizado foi o modelo CH8568 da CBN e tem velocidade de conexão 40 Mbps.

Serão utilizados softwares desenvolvidos para sistemas Linux a fim de realizar o ataque e, no *host* que receberá o ataque será utilizado um sistema operacional e um programa para monitorar a rede.

Nas simulações que serão feitas, utilizaremos os sistemas operacionais Kali Linux, como sistema atacante, e Windows 8, como vítima, configuradas numa mesma rede local com os IP's 192.168.0.15 e 192.168.0.249, respectivamente.

Para a realização dessas simulações em laboratório — que podem ser reproduzidas em ambiente real—, utilizaremos os seguintes sistemas operacionais e softwares:

1. SO Kali Linux, disponível em https://www.kali.org/get-kali/;
2. SO Windows 8;
3. Software Wireshark para Windows, analisador de tráfego de rede, disponivel em https://www.wireshark.org/download.html;
4. Software Scapy para Linux, disponível em http://www.secdev.org/projects/scapy/.
5. Software Hiping3 para Linux, disponivel em https://www.kali.org/tools/hping3/.

4.1 Simulação de ataque SMURF

Foi realizado o ataque *Smurf*, que é um ataque com grande potencial devido a sua

utilização da rede como forma de ampliar o ataque. Um ataque desse tipo é descrito como distribuído no qual um invasor tenta inundar um servidor com pacotes do *Internet Control Message Protocol* (ICMP). Assim que são feitas solicitações com o endereço IP falsificado do alvo para uma ou mais redes de computadores, essas redes encaminham para o servidor atacado, amplificando o tráfego de ataque inicial possivelmente sobrecarregando o alvo, deixando-o sem acesso. [Bardal, 2014]

Usando a ferramenta scapy no atacante, que enviará pacotes ICMP para a vítima, inundando a rede. Os comandos utilizados foram:

sudo scapy

send(IP(dst="192.168.0.249",src="192.168.0.15")/ICMP(),count=1000,verbose= 1)

Onde ‘src’ e ‘dst’ identificam, respectivamente, os IPs do atacante e vítima; count é utilizado para definir o número de pacotes enviados (Figura 1).

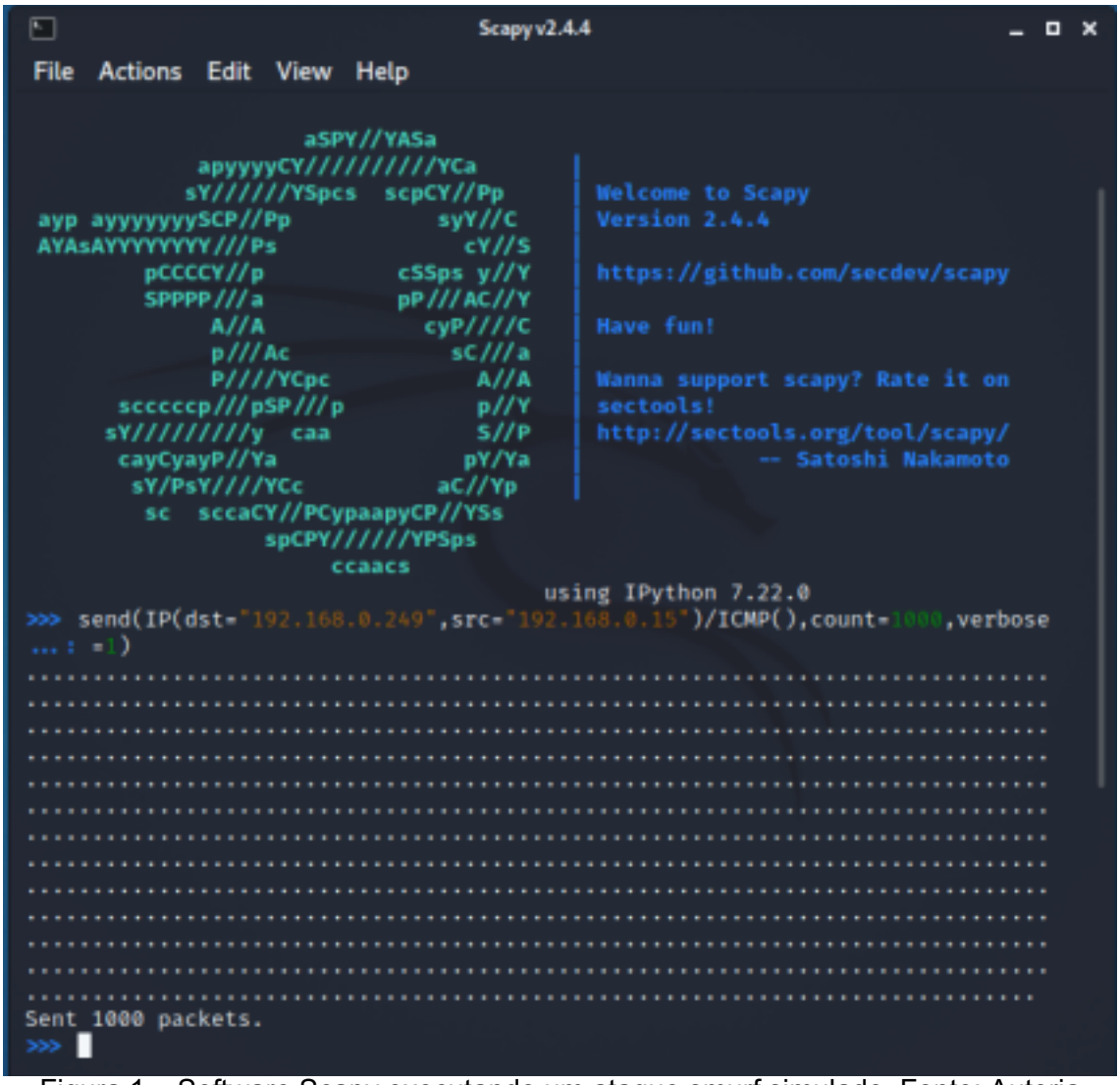

Figura 1 – Software Scapy executando um ataque smurf simulado. Fonte: Autoria própria.

É possível ver que os pacotes ICMP Request sendo respondidos ao atacante com ICMP Reply através do Wireshark (Figura 2)

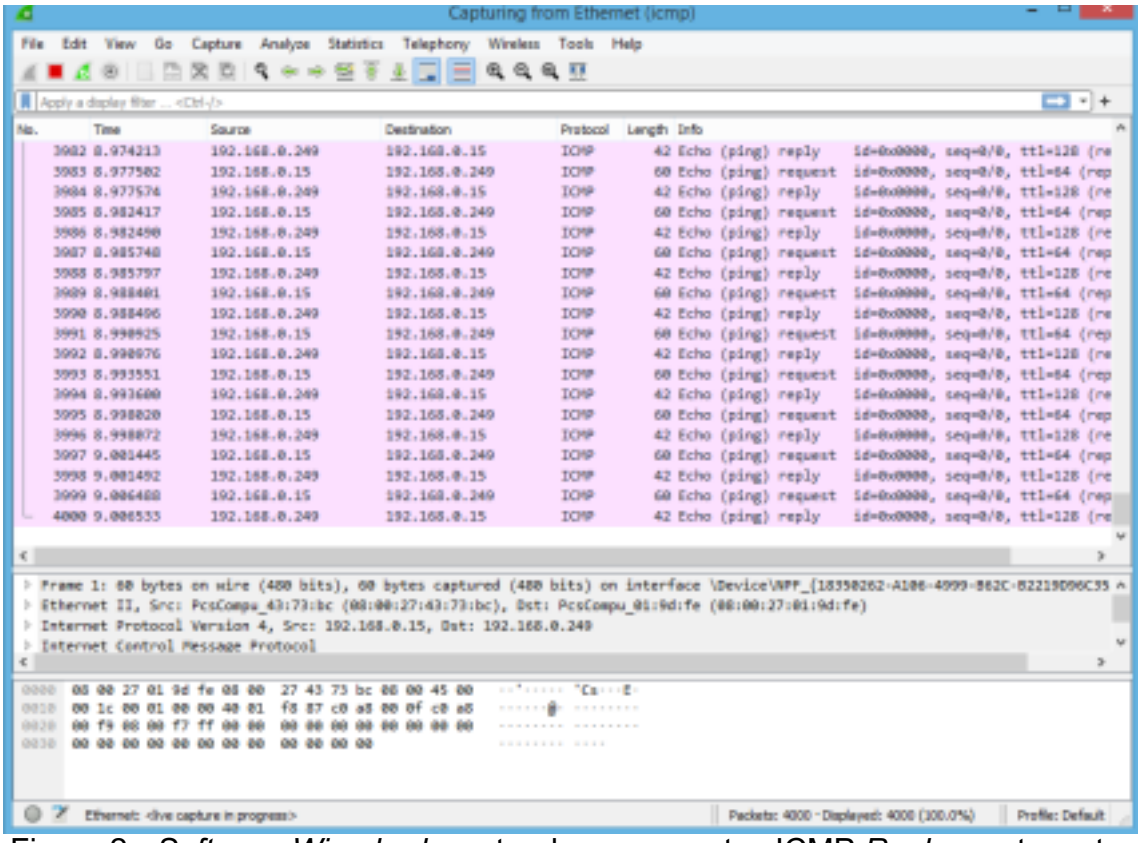

Figura 2 - *Software Wireshark* captando as respostas ICMP *Reply* ao atacante.
Fonte: Autoria própria.

Através do gerenciador de tarefas do *Windows* também foi possível verificar o aumento de consumo da interface de rede (Figura 3), corroborando que é um ambiente virtual, portanto, não houve o consumo total da mesma; sendo assim, em um ataque real, em que a ocupação seria total.

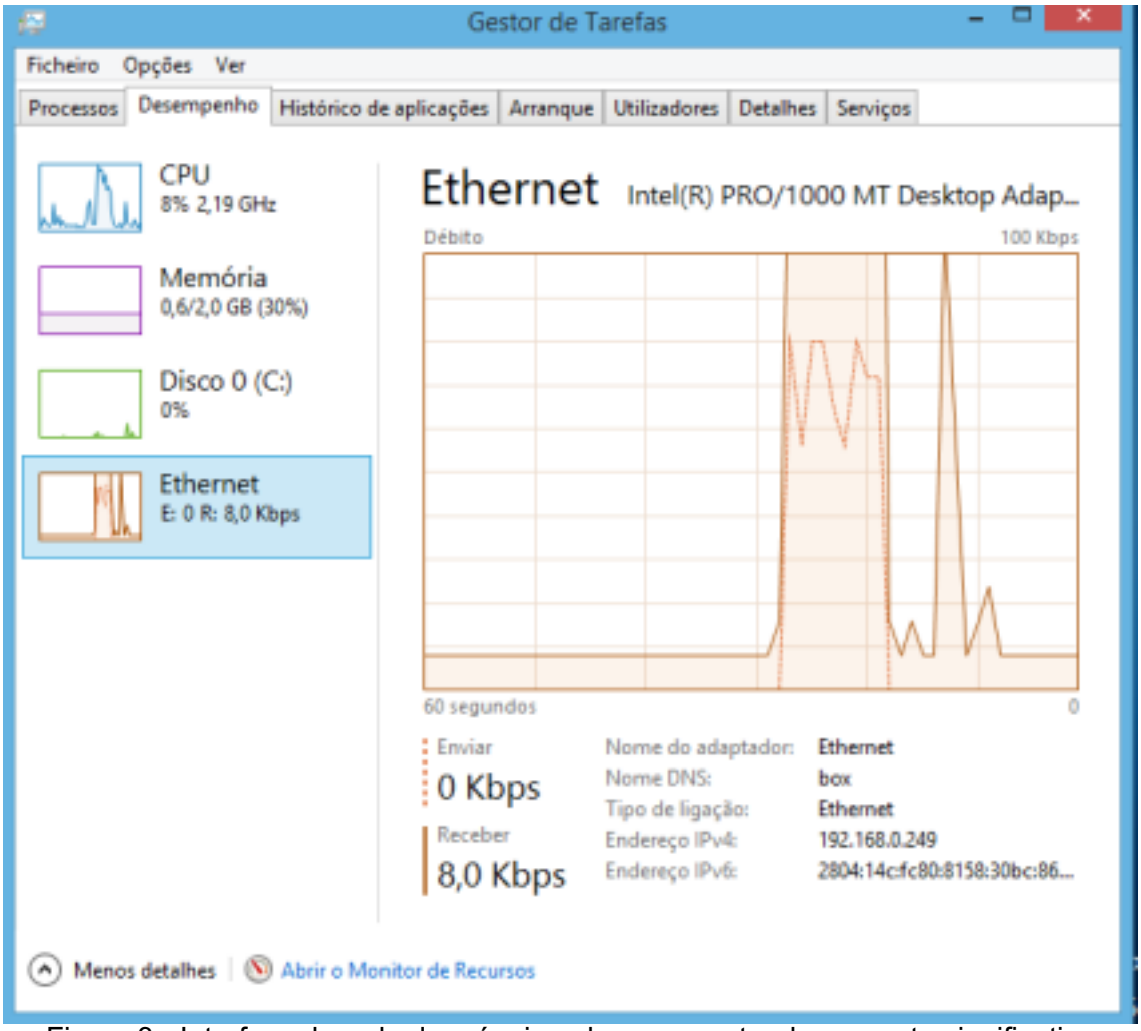

Figura 3 - Interface de rede da máquina alvo apresentando aumento significativo durante ataque *smurf.* Fonte: Autoria própria.

Vemos antes do ataque, a leitura da interface de rede está em 8kbps, durante o ataque a máquina atinge 100Kbps e, assim que o ataque é cessado, a leitura volta a 8Kbps.

4.2 Simulação ataque SYN Flood

Uma inundação de SYN é um ataque de negação de serviço (DDoS) projetado para tornar um servidor inacessível ao tráfego legítimo inundando a fila de conexões, indefinidamente e, decorrente a isso, consumindo todos os recursos disponíveis, ao sobrecarregar todas as portas disponíveis [Bardal, 2014].

Neste ataque foi utilizado o *hping3*, que é um gerador e analisador de pacotes de código aberto para o protocolo TCP/IP. Os comandos utilizados foram:

sudo hping3

hping3 -i u1 -S -p 80 192.168.0.249

Os parâmetros -i u1 e -S identificam, respectivamente, intervalo de 1ms entre as mensagens e o uso do TCP SYN, e a porta (HTTP). utilizada é identificada por -p 80.

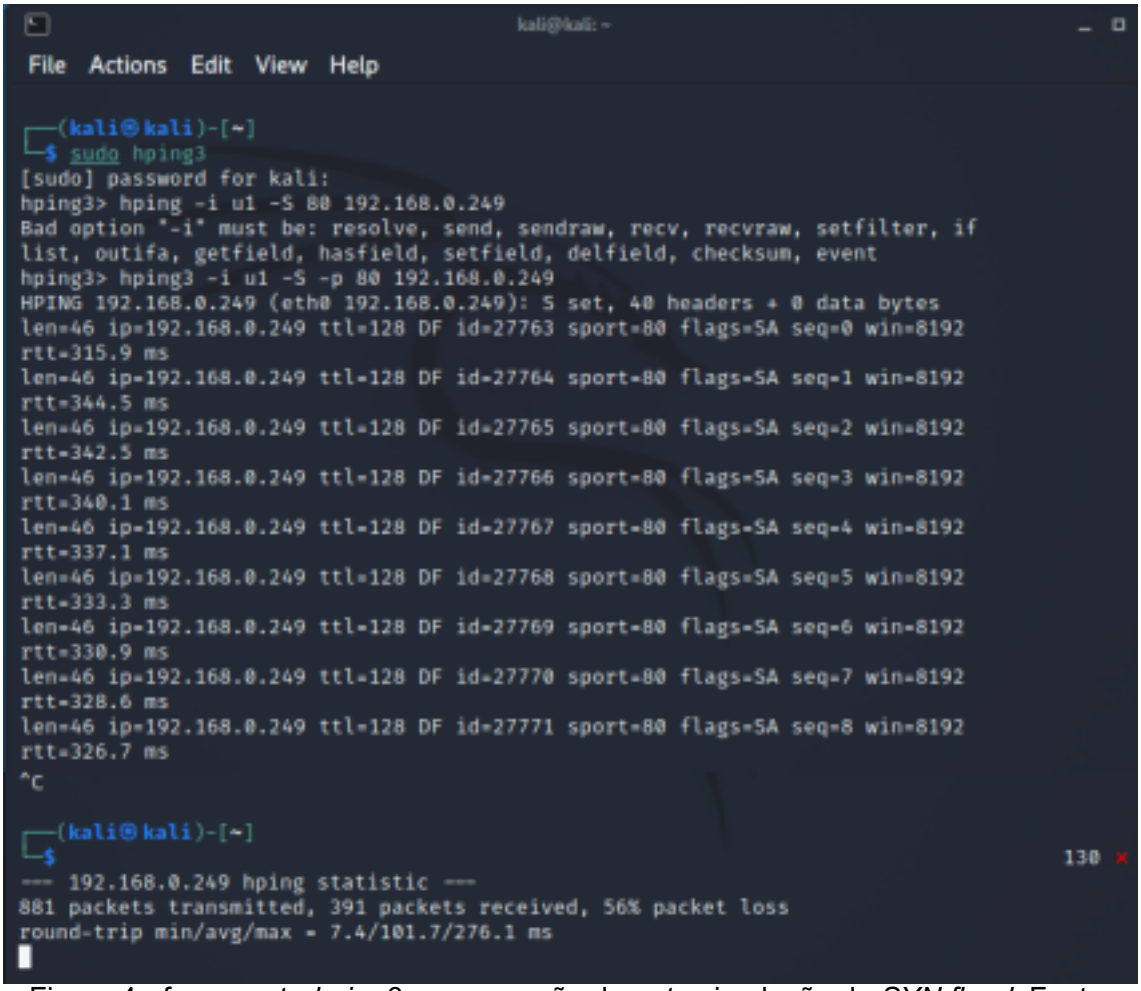

Figura 4 - ferramenta *hping3* em execução durante simulação de *SYN flood.* Fonte: Autoria própria.

Durante o ataque, foi observado, através do Wireshark na máquina alvo, o grande número de pacotes enviados (881 transmitidos, 391 recebidos), como podemos ver na Figura 5.

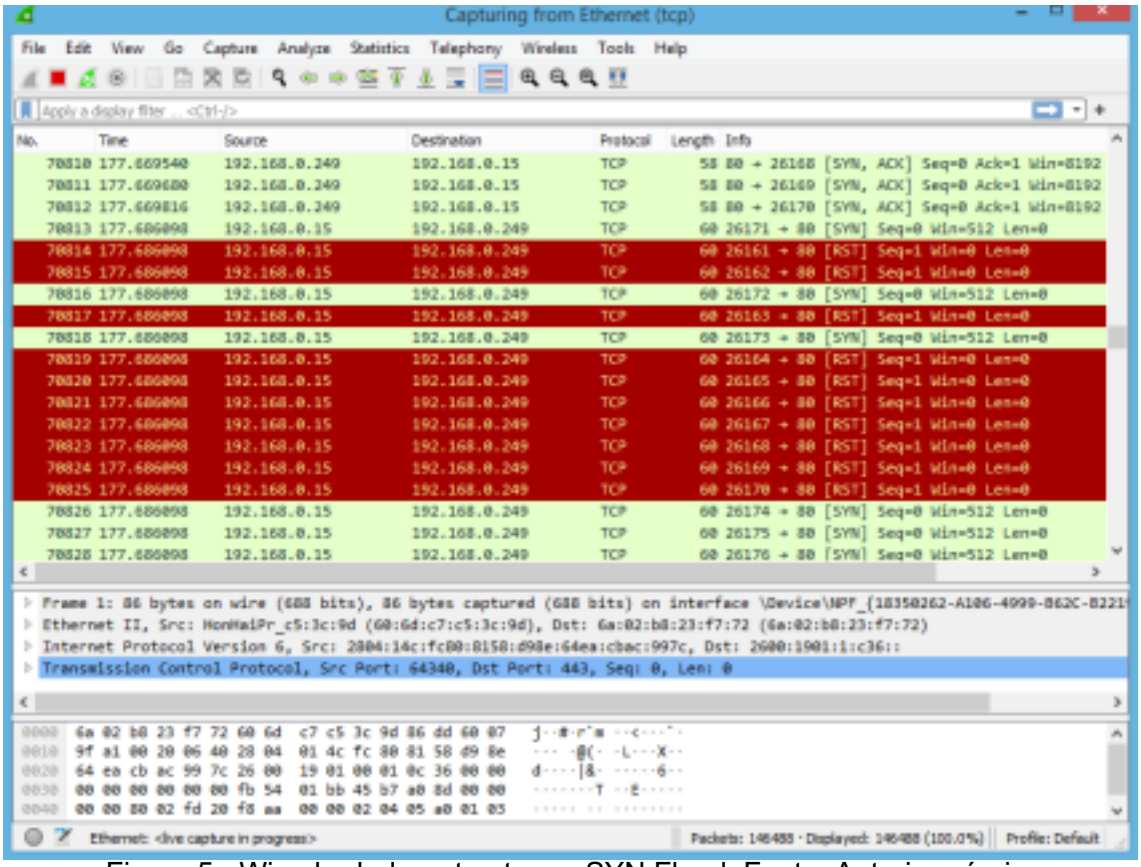

Figura 5 - Wireshark durante ataque SYN Flood. Fonte: Autoria própria.

Durante o ataque a CPU chegou a atingir 100% de sua capacidade, inutilizando a máquina, e o ataque teve que ser interrompido para que a máquina pudesse voltar a ser utilizável. (Figura 6)

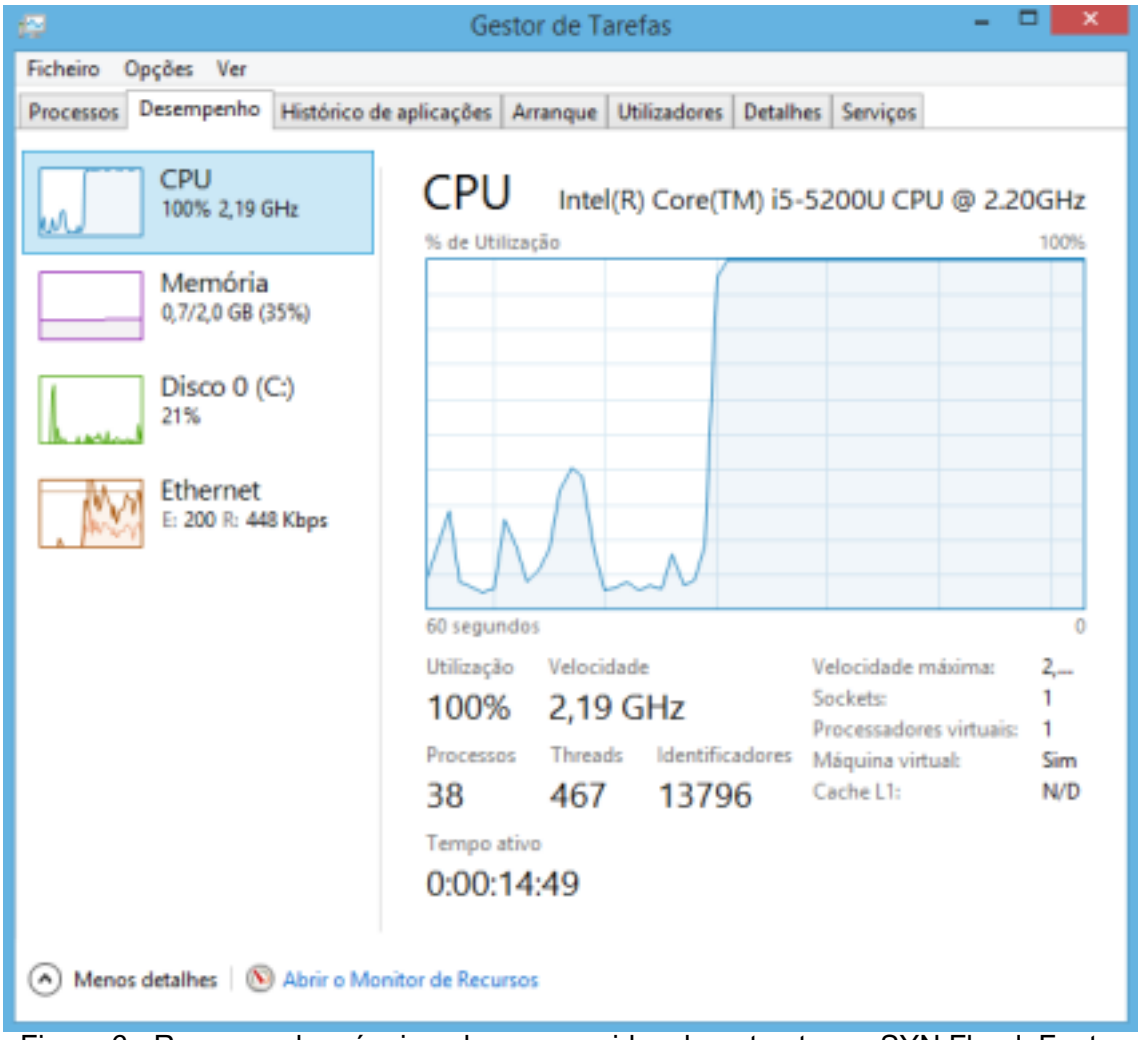

Figura 6 - Recursos da máquina alvo consumidos durante ataque SYN Flood. Fonte: Autoria própria.

A utilização da CPU chega a sua utilização máxima durante o ataque, e, na interface de rede, é possível ver que antes do ataque, recebia 8 Kbps e, durante o ataque, chega a, aproximadamente 712 Kbps.

5 Resultados

Através dos testes foi possível ver em tempo real como um ataque de negação de serviço age, esgotando os recursos da máquina alvo, mostrando que os sistemas são passíveis de sofrer com esse tipo de ataque.

Os resultados dos testes fornecem uma melhor compreensão de como ataques de negação de serviço impactam e abrem possibilidades para pesquisas futuras, como métodos para identificar e impedir os possíveis futuros ataques.

É válido salientar que a análise foi feita baseada em cima da virtualização;

Dessa forma, como toda virtualização apresenta desvantagens em relação ao modelo físico, é possível que os resultados apresentem disparidades em relação a um ataque real. Porém, o objetivo da pesquisa não era comparar com uma situação real, e sim mostrar o ataque acontecendo e demonstrar didaticamente um passo a passo. Para tanto, a virtualização se mostrou eficiente.

A fim de auxiliar na visualização do ataque em tempo real foram gravadas em vídeo as duas simulações, que está disponível em https://youtu.be/AUgIV9nCjYc.

6 Conclusão

Com esse trabalho foi possível ver que a utilização de um ambiente virtualizado é vantajosa por ser mais fácil aliar as atividades teóricas às práticas, tornando mais didático o aprendizado, além de propor um ambiente para realizar os testes e coletar métricas em sistemas atualmente utilizados.

É importante citar também o fator legal: como esse tipo de prática não pode ser feita (nem a fins de estudo) sem a autorização do alvo, a virtualização se torna um facilitador na hora de montar um laboratório, já que ter um ambiente com várias máquinas nem sempre é possível, fora a economia de energia e redução de espaços físicos.

Trabalhos futuros podem agregar ainda mais ao resultado desta pesquisa tratando da parte de detecção e impedimento de possíveis ataques, utilizando um firewall - como, por exemplo o *PfSense*-, e até mesmo replicar este mesmo cenário em ambientes mais atuais e em aparelhos móveis, tendo em vista sua ampla utilização atualmente.

7 Referências

BCP:Portal de boas práticas para internet no Brasil, Antispoofing. Disponível em: <https://bcp.nic.br/antispoofing>. Acesso em: 16 jan. 2022.

Broad, J; Bindner, A. Hacking com Kali Linux: Técnicas práticas para testes de invasão. Novatec, fev./2015.

Constantin, L. DDoS attacks increase in number, endanger small organizations, 2015. Disponível em <https://www.pcworld.com/article/3012963/ddos-attacks-increase-in-number-end anger-small-organizations.html>. Acesso em: 21 dez. 2021.

Donato, H; Donato, M. Etapas na Condução de uma Revisão Sistemática, Revista Científica da Ordem dos Médicos, mar/2019

Engebreston, P. The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy. Syngress, ago./2013

GOMES, L; Araújo, M; Campos, V. Negação de Serviço e Botnets, UFRJ, p. 1-17, 10 jan. 2015. Disponível em: https://www.gta.ufrj.br/grad/15_1/dos/index.html.

Maciel, R. Avaliação do impacto de ataques DDoS e Malware: uma abordagem baseada em árvore de ataque. ATTENA, ,Pernambuco, v. 1, n. 1, p. 8-91, ago./2018.

Menascé, D.: Virtualization: Concepts, Applications, and Performance Modeling. Int. CMG Conference 2005: 407-414.

Monteverde, W. Estudo e Análise de Vulnerabilidades Web. 2014. 71 f. TCC (Graduação)—Curso de Sistemas para Internet,
Universidade Tecnológica Federal do Paraná, Campo Mourão—PR, 2014.

Santos, C, Pimenta, C e Nobre, M; The PICO strategy for the research question construction and evidence search. Revista Latino-Americana de Enfermagem [online]. 2007, v. 15, n. 3 [Acessado 18 Junho 2021] , pp. 508-511. Disponível

em: <https://doi.org/10.1590/S0104-11692007000300023>

Weidman, G. Testes de invasão: uma introdução prática ao hacking. São Paulo—SP: Novatec, abril/2015.

Zuben, M. Recomendações para Melhorar o Cenário de Ataques Distribuídos de Negação de Serviço (DDoS). 2016.