Teste do Framework Cybergrenade em Ambiente Corporativo

Jafar Mohammed Untar[1], Daves Marcio Silva Martins[2], Sandro Roberto Fernandes[3]

[1 2 3] Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais (IF Sudeste MG)

Juiz de Fora, MG - Brasil

jafar.untar@gmail.com, daves.martins@ifsudestemg.edu.br, sandro.fernandes@ifsudestemg.edu.br

Abstract. *With the increase of the world's information flow and internet use, the need of an information security professional has made it even more necessary to guarantee the confidentiality, integrity and availability of these data. The pentesting or penetration testing is important to assess the system security by the look of an attacker. As this stage demands time and a lot of manual action by the professional, the opportunity of automation emerged. The Cybergrenade framework found during the research appears to serve this purpose. This paper tested it but now on a more enterprise realistic environment, simulated using 3 virtual machines, and it shows that the tool efficacy limits itself on unprotected networks, since a firewall installation blocked the breach from succeeding.*

Resumo. *Com o aumento do fluxo de informações e do uso da internet no mundo, o profissional de segurança da informação se faz cada vez mais necessário para garantir a confidencialidade, integridade e disponibilidade desses dados. O processo de pentesting ou teste de intrusão é importante para que se avalie a segurança de um sistema do ponto de vista de um atacante. Como essa etapa demanda tempo e muita ação manual do profissional, surgiu a oportunidade de automatização. O framework Cybergrenade aparenta cumprir essa proposta. Este trabalho o testou em um ambiente mais próximo da realidade de um ambiente organizacional, simulado utilizando 3 máquinas virtuais, mostrando que a eficácia da ferramenta se limita a redes desprotegidas, visto que a instalação de um firewall na porta da rede impediu a invasão de ser bem sucedida.*

1. Introdução:

Com um mundo cada vez mais conectado à internet, o Brasil já possui mais de 82,7% dos domicílios com acesso à internet [IBGE, 2019]. Um estudo feito pela plataforma Cupom Válido, mostrou mais de 150 milhões de usuários brasileiros em redes sociais, além do que, só em 2020, mais de 51% das transações bancárias foram feitas digitalmente pelo celular [FEBRABAN, 2021], evidenciando que a vida financeira das pessoas já tem forte presença no mundo virtual. Com um grande fluxo de dados e informações sensíveis como números de cartão de crédito, senhas de banco e de contas bancárias que passam pelas redes, a necessidade de se assegurar e proteger dados sigilosos e pessoais de acesso não autorizado está em constante crescente.

A área de cibersegurança, a prática que protege computadores e servidores, dispositivos móveis, sistemas eletrônicos, redes e dados contra ataques maliciosos, também chamada de segurança da tecnologia da informação ou segurança de informações eletrônicas [KASPERSKY, 2022], vem recebendo cada vez mais atenção e estima-se um total de US$ 1 bilhão investidos no país em 2022 [Ramos, Luciano, 2022].

Um dos principais motivos para essa atenção se deve ao aumento das complexidades de cibersegurança, devido às diferentes culturas e processos organizacionais, e nas dificuldades para encontrar e manter profissionais qualificados, corroborando em 40% das empresas consultadas pela IDC Brasil afirmarem que a falta de especialistas nas equipes é um fator crítico [Ramos, Luciano, 2022]. Um fator que justifica essa precariedade de profissionais capacitados se deve ao alto nível de conhecimento necessário para a função.

Além disso, um levantamento feito pela *Fortinet Threat Intelligence Insider Latin America*, mostrou que em 2021 o Brasil sofreu mais de 88,5 bilhões de ataques cibernéticos desde o começo da pandemia do COVID-19. Com um aumento de mais de 950% nesse número de crimes virtuais, colocou o país no segundo lugar entre os que mais receberam ataques na área da América Latina e Caribe, aumentando em 40% a procura das empresas por seguros cibernéticos só no primeiro semestre de 2022 [CNN, 2022].

O profissional da cibersegurança deve conhecer com profunda extensão o funcionamento de um sistema e de uma rede para que se possa avaliar e definir pontos de proteção necessários, com possibilidades de trabalhar em diversas áreas dentro desse próprio ramo da tecnologia. Dentre várias especializações, uma delas é a segurança ofensiva, responsável por invadir um sistema e assim descobrir suas possíveis vulnerabilidades, documentar e apresentar a empresa para que possam melhorar seu sistema [Macedo, Iran, 2019].

Nas etapas de trabalho do profissional de segurança ofensiva, é a etapa de *pentesting*, *penetration testing* ou teste de intrusão, etapa que se executam diversos testes e técnicas a fim de invadir um sistema e a partir dali se descobrir até onde se pode chegar, buscando atingir algum recurso valioso e de alta confidencialidade para uma empresa. O *pentesting* demanda do profissional tempo de análise e trabalho manual para que se possa fazer o teste de invasão em diversos pontos que podem ser vulneráveis, indicando a possibilidade de erro humano em uma demanda minuciosa e com muitos detalhes abstratos como aproveitar uma falha em modelo de negócio, e que, de acordo com um levantamento feito pelo site da Redscan em 2021, mais de 50 vulnerabilidades são reportadas todos os dias. Assim, surge uma oportunidade de automatização dessa parte do processo para aumentar a eficiência do *Pentest,* com maior cobertura de testes e diminuição do tempo de execução.

2. Trabalhos Relacionados

Em Anurag Akkiraju et al (2017), existe a proposta de *pentesting* em redes locais utilizando-se de *SBC [Single Board Computers]* como por exemplo um *Raspberry Pi*, computadores de uma só placa. O autor utilizou-se de ferramentas já instaladas na distribuição Linux *Kali* em um *script* feito em *Python* para automatizar o processo de teste de invasão, desde o scan da rede até o *exploit*. O *framework* intitulado *Cybergrenade* se mostrou possível de ser executado nos *SBCs* sem dificuldade, conseguindo atingir as

máquinas que apresentavam vulnerabilidades, sendo o experimento feito em 3 máquinas virtuais, com o parâmetro de sucesso sendo a obtenção de uma *shell* na máquina alvo.

Em Ankur Chowdary et al (2020), a proposta para solução do problema é a criação de um algoritmo de aprendizado de reforço baseado na Deep-Q Network [DQN] para identificar, por meio de grafos, possíveis caminhos para planos de ataque em uma rede, e o chamam de Autonomous Security Analysis and Penetration Testing Framework [ASAP] que em tradução livre fica Framework de Automação de Análise de Segurança e Testes de Intrusão. Para gerar os grafos de ataque, o algoritmo primeiro necessita das informações da configuração de rede e das informações das vulnerabilidades, e utilizando da mudança de estados do atacante para planejar o melhor caminho de ataque com ajuda da aprendizagem por reforço do framework. O ASAP conseguiu escalar bem em redes maiores e conseguiu generalizar mesmo em diferentes cenários, possibilitando assim o atacante a simplesmente executar o plano de ataque gerado pela ferramenta.

Em S.P Kadam et al (2016), se apresentam diversas formas de se atacar uma rede wireless, demonstrando durante o texto as técnicas e suas nomenclaturas. O autor também explica como diversas ferramentas podem ser combinadas, corroborando na criação de um aplicativo para a plataforma Android para utilizar-se de recursos como interface gráfica e a mobilidade de um smartphone. Assim, foi criada essa ferramenta que possui várias outras embutidas, e que tirou a necessidade de digitar comandos tornando o processo todo interativo por meio de menus, facilitando também para quem não possui o conhecimento técnico. Também é dito que a conveniência de poder executar esse processo por meio de um smartphone aumenta a furtividade ao entrar em alguma empresa para realizar o pentest, além do fato que as possibilidades com o telefone são maiores devido a suas conectividades 3G e 4G, possibilitando assim testar via diferentes formas de conexão.

Em Tian-yang Zhou et al (2019) também é mostrado uma resolução do problema apresentado utilizando inteligência artificial muito parecido com o apresentado por Ankur Chowdary, com a utilização de aprendizagem por reforço para criar caminhos de ataque com o *Deep-Q Network*, surge o *NIG-AP Framework*. Com as métricas de recompensa do algoritmo voltadas para o *Score CVSS [Common Vulnerability Scoring System]*, uma medida que mostra o impacto de cada vulnerabilidade, o *framework* consegue fazer um plano de invasão de acordo com os caminhos com possíveis maiores impactos. Assim como o artigo acima, a validação dos grafos é feita utilizando-se de ações do *Metasploit Framework* para a construção da linha de *exploits* que serão utilizados no fluxo do *pentest*.

Em Yaroslav Stefinko et al (2016), é mostrado um comparativo entre o processo de *pentest* manual e o automatizado, além de mostrar algumas metodologias de *pentesting* como o *NIST* e o *BackTrack*. Com a comparação de que muitas vezes, o processo manual pode ser mais utilizado por conta de sua popularidade e eficácia, visto a grande diversidade de falhas e suas peculiaridades, o processo automatizado pode facilitar o andamento de forma que diminua o tempo necessário para conclusão do teste. Porém, também é mostrado que muitas vezes as ferramentas com essa finalidade demandam um treinamento também por parte da equipe, e que para maior abrangência de falhas encontradas e uma maior aplicabilidade, a execução e customização deve ser feita por um profissional já experiente.

Visto uma lacuna que poderia ser preenchida no trabalho de Anurag, sendo ela a possibilidade de explorar o uso do *Cybergrenade* em um ambiente mais próximo da

realidade do meio empresarial, pois o artigo original não cobre testes com alguma camada de segurança a mais, este trabalho visou testar o algoritmo nessas situações, para que se possa avaliar a relevância desse algoritmo para sua utilização futura tanto no meio profissional quanto acadêmico.

3. Fundamentação Teórica:

3.1 Firewall e Redes Empresariais

De acordo com o site do Microservice, o *firewall* é a "primeira linha de defesa" dos dispositivos, e é uma ferramenta em *software* ou *hardware* responsável por fazer a filtragem dos dados que passam pela rede [empresarial ou doméstica] com base em um conjunto de regras e instruções de segurança pré-estabelecidas, servindo como uma barreira de proteção. Uma empresa normalmente possui uma rede interna para utilização dos funcionários, e com ela existe a necessidade de validação de acessos, defesa de recursos, monitoramento de tráfego, registro e reporte de incidentes. Assim, para a proteção de dados e conteúdo sensível de um sistema organizacional, a existência do *firewall* é justificada.

3.2 Dual-Homed Firewall:

De acordo com Sérgio Duarte [2002], um computador entre duas redes, que neste trabalho foi definido pela máquina virtual *OPNSense*, pode servir como *Dual-Homed Firewall*, que é uma forma de implementação de *firewall* onde duas interfaces de rede são utilizadas e o roteamento livre é desativado ou seja, não se pode fazer a transmissão direto de pacotes entre a rede externa e a rede interna. Deste modo, os sistemas entre as duas redes devem passar por um gateway central que efetua a filtragem de pacotes e nega o estabelecimento de conexões com serviços não autorizados, definidos na configuração, que conta como uma desvantagem ser complexa de início.

3.3 Kali Linux:

Apresentado primariamente como um sucessor ao projeto *BackTrack Linux* criado pela empresa americana fornecedora de serviços de *pentesting* e forense digital, a *Offensive Security*, o *Kali* é uma distribuição *Linux* com diversas ferramentas de testes de segurança já instaladas, fornecendo assim um ambiente adequado para as ações do profissional de segurança, como também funcionalidades de personalização do kernel do sistema, auto-conexão em *VPN [Virtual Private Network* - Rede Virtual Privada*]* reversa, opções de instalação tanto em *smartphones* por meio do *Kali NetHunter*, como também por meio de conteinerização, entre outros [KALI ORG, 2022]. A distribuição desse sistema *Linux* já se mostrou relevante no mundo da cibersegurança visto sua popularidade, com mais de vinte e cinco mil resultados por seu nome no *Google Acadêmico*.

3.4 Common Vulnerabilities and Exposures - CVE

O programa *CVE* é uma iniciativa que identifica, define e cataloga vulnerabilidades de cibersegurança, enumerando as falhas encontradas pelas empresas que participam do programa. O *CVE* padroniza e, por meio de números, permite que os profissionais saibam de qual vulnerabilidade estão falando e como mitigar, já que no site é possível a visualização da descrição da falha e muitas vezes informações de *PoCs [Proof of Concept* - Prova de Conceito*]* fornecidas por parceiros que reportaram a falha, permitindo assim maior entendimento na causa e priorizar a resolução do problema.

Juntamente com o *CVSS [Common Vulnerability Scoring System]*, uma medida para calcular em uma escala de 0 a 10 a gravidade de cada *CVE*, agrupando por métricas de impactos Base, Temporal e Ambiental [NVD NIST, 2022], é possível fazer uma base de dados para organização de um resultado de teste de invasão e endereçamento de ameaças que podem permear um sistema.

3.5 Metodologia de Pentesting:

Existem diversas metodologias que podem guiar um processo de *pentest*, de forma que o profissional possa ter um caminho como base para sua execução, com diferenciação muitas vezes por algumas etapas que são incluídas ou não por metodologia, mas que serviram no planejamento do seguimento deste trabalho de forma que justificassem cada decisão. São elas, por exemplo:

- *Open Source Security Testing Methodology Manual - OSSTMM 3*, ou Manual da Metodologia Testes de Segurança de Código Aberto é uma metodologia de teste voltada para locais físicos, interações humanas e todas as formas de comunicação, seja via *wireless*, cabeada, analógica ou digital [ISECOM, 2010]
- *Open Web Application Security Project - OWASP*, ou Projeto Aberto de Segurança de Aplicações Web, é uma fundação sem fins lucrativos com projetos abertos à comunidade, sendo alguns deles metodologias de testes de intrusão voltados para proteção de aplicações *web* e sistemas mobile. A *OWASP* possui um projeto nomeado de *OWASP Top Ten*, que divulga todo ano uma lista com os dez riscos de segurança que mais ocorreram e foram de mais relevância no escopo de aplicações na internet [OWASP, 2022].
- *National Institute of Standards and Technology - NIST*, uma organização americana de padrões e que é comumente um requisito mandatório para empresas seguirem e ficarem de acordo com as normas do país. O manual do *NIST* possui ênfase no aprimoramento geral de cibersegurança de uma organização, com sua versão mais recente sendo voltada a infraestrutura crítica de cibersegurança. Além disso, sua metodologia de *pentesting* guia os estabelecimentos que desejam ficar de acordo com a norma, com obrigações de controle de cibersegurança e mitigação de riscos que o negócio deve seguir [VUMETRIC, 2019].
- *Penetration Testing Methodologies and Standards - PTES Framework*, uma metodologia que foca em coletar o máximo de informações e inteligência acerca do ambiente para maior entendimento dos valores e recursos que devem ser protegidos e explorados, e assim se possa iniciar o teste de intrusão com maior coerência em cada tomada de decisão [PENTEST STANDARD, 2014].

3.6 Defesa em Profundidade:

Inicialmente um conceito militar, o conceito de defesa em profundidade ou *Defense in Depth*, em inglês, de acordo com o site da Fortinet, é uma estratégia adotada na criação de sistemas de segurança de forma que existam várias camadas de proteção na infraestrutura dos recursos de uma organização.

O pensamento deve ser aplicado tanto de forma técnica quanto pessoal, com treinamento de pessoas e instalação de recursos de hardware que auxiliam no processo de amenização de danos, para que em caso de incidente e de violação de segurança sua

extensão seja a menor possível e os ativos mais importantes de uma empresa sejam nada ou minimamente atingidos.

Com este pensamento, o objetivo de entender este conceito levou a apresentar a proposta de incrementar mais uma camada de segurança neste artigo, que no caso, a presença de um *firewall* se mostrou adequada para o intuito de proteger os servidores que se encontram na rede.

3.7 Exploit

Diante de uma brecha no sistema, é necessário especificar de que forma é possível o aproveitamento ou utilização deste meio para invasão. Um programa ou pedaço de código utilizado para este fim, encontrados normalmente para fins maliciosos, é chamado de *exploit* [CISCO]. Os *exploits* podem ser encontrados publicamente em sites como *Exploit-DB* que possui um grande banco de dados com provas de conceito possíveis de aplicação em sistemas reais para o *pentest*, permitindo que empresas possam corrigir a falha analisando diretamente o código executado.

4. Metodologia:

A metodologia deste trabalho consiste em executar o *script* em uma máquina que representa um atacante ou um profissional de segurança executando um *pentesting*. Assim, foi pensado em montar um laboratório com uma máquina atacante, uma máquina que será responsável por ser o *firewall*, e duas máquinas vulneráveis com vulnerabilidades conhecidas. A execução dos testes foi dividida em duas etapas, sendo a primeira o teste sem a implementação do *firewall*, e a segunda com a filtragem de pacotes do dispositivo habilitada.

O Cybergrenade é um framework de um único script feito para ser executado em máquinas SBC [Single Board Computers]. O projeto utilizou como ambiente para execução o Kali Linux, poupando assim o usuário da instalação de todas as ferramentas necessárias. A execução do script se inicia assim que o computador é conectado com a rede que se deseja executar o teste. Para isso, o script se utilizou da linguagem Python que possui comandos que têm integração direta com o terminal do sistema, possibilitando a execução das ferramentas, que no caso seria a utilização do nmap, uma ferramenta para scan de redes, o Metasploit Framework, que possui uma grande base de scripts e que executa o exploit para determinadas vulnerabilidades, além de ser um dos frameworks de pentesting mais utilizados no mundo de acordo com o blog Software Testing Help, e o OpenVAS, um scanner automatizado de vulnerabilidades de código aberto.

Durante a revisão bibliográfica era esperado encontrar o *framework* já pronto para implementar e executar os devidos testes no laboratório, porém não foi encontrado nenhuma fonte em que houvesse o código aberto, criando a necessidade de recriar manualmente o script, e utilizando o artigo original como base para seguir o fluxo de execução do programa, como visto na Figura 01 mantendo os utilitários que existem no próprio sistema operacional do *Kali*, como por exemplo o *nmap*, um escaneador de redes, no código produzido.

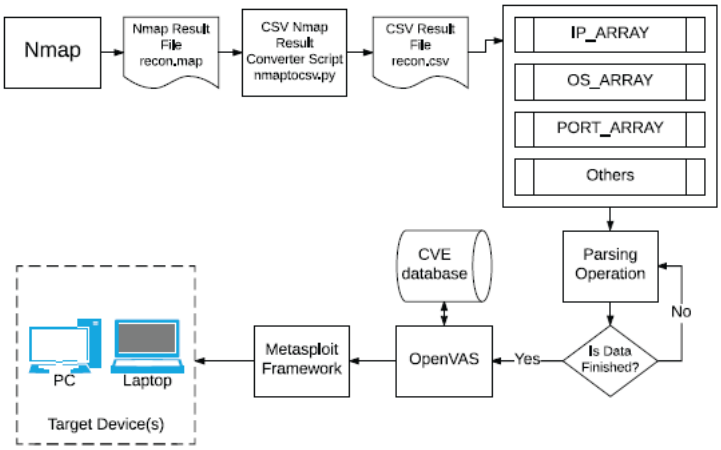

Figura 1. Fluxo de execução do Cybergrenade [Fonte: Anurag Akkiraju]

Na implementação original do *framework*, é dito que para a execução de tarefas no scanner de vulnerabilidades se deveria enviar um *XML* como parâmetro de um comando do *Linux* do próprio *OpenVAS*, porém, visto a existência de uma biblioteca do *Python* que lida diretamente com a aplicação, simplificou a integração do scanner com métodos que fazem as funções, necessárias para o teste, na ferramenta.

4.2 Criação do Laboratório:

O artigo original do *Cybergrenade* apresentou um teste somente em duas fases, sendo uma fase com somente uma máquina vulnerável e outra fase com testes em três máquinas vulneráveis na rede, utilizando-se do recurso de virtualização com o programa *VirtualBox VM* da Oracle. Porém, pesquisando por "*Network Diagram*" e "*Enterprise Network Architecture*" se encontram diversos resultados exemplificando modelos de rede que demonstram, a alto nível, a disposição de computadores e dispositivos em uma rede empresarial, que difere da rede apresentada pelo autor original. Ao observar os diagramas, se observa a presença de um *firewall* antes da entrada na rede principal. Também se pode observar a existência de um *switch* de rede para que se conecte todos os aparelhos de forma que se tornem acessíveis para os computadores internos, como por exemplo impressoras e servidores de arquivos.

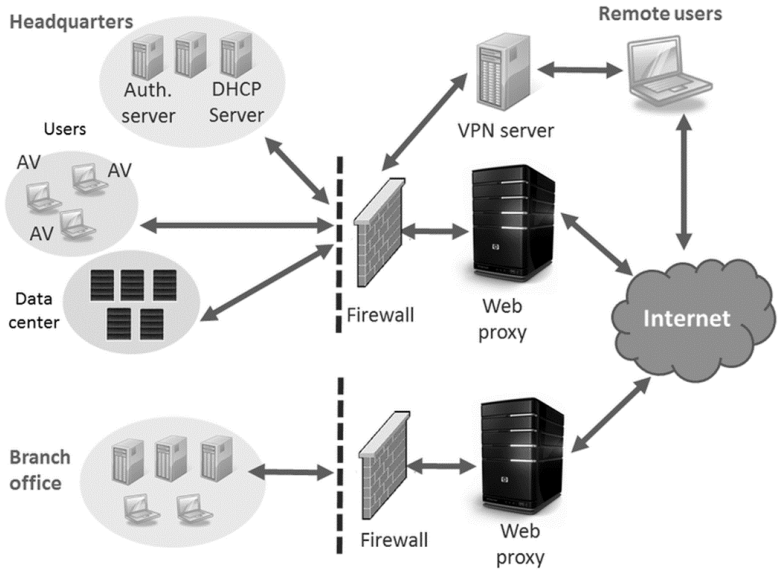

Figura 2. Exemplo de uma rede empresarial [Fonte: Zhou Li]

Com isso, se planejou a criação de um laboratório virtual utilizando-se das mesmas tecnologias no artigo original, porém agora com a inserção de um *firewall* responsável por filtrar as requisições e o tráfego de rede que acontece entre as máquinas e a internet. Para isso, utilizando a solução *OPNSense*, um *firewall* de código-aberto, se montou um laboratório que consiste em 2 máquinas dentro de uma rede isolada protegida por esse *firewall*.

Para a instalação do *firewall*, se criou uma nova máquina virtual no *VirtualBox*, com as configurações de *BSD* e *OpenBSD [64-bit],* um sistema gratuito com licença *Berkeley Software Distribution [BSD]*, baseado em *UNIX* [OPENBSD, 2022], essa máquina será responsável por atuar como *firewall* e *gateway* para a rede privada que ela circunda. Assim deve-se criar adaptadores de rede no modo *Host-Only Adapter* para que se separe em uma rede interna, e alocar a máquina atacante [*Kali Linux*] em uma rede separada para que se simule um atacante externo como mostrado na Figura 3. Todas as máquinas-alvo devem ser configuradas de modo que utilizem do mesmo adaptador *Host-Only* que a máquina do *OPNSense*, para que todos os pacotes de rede passem primeiramente por ela como visto na Figura 3.

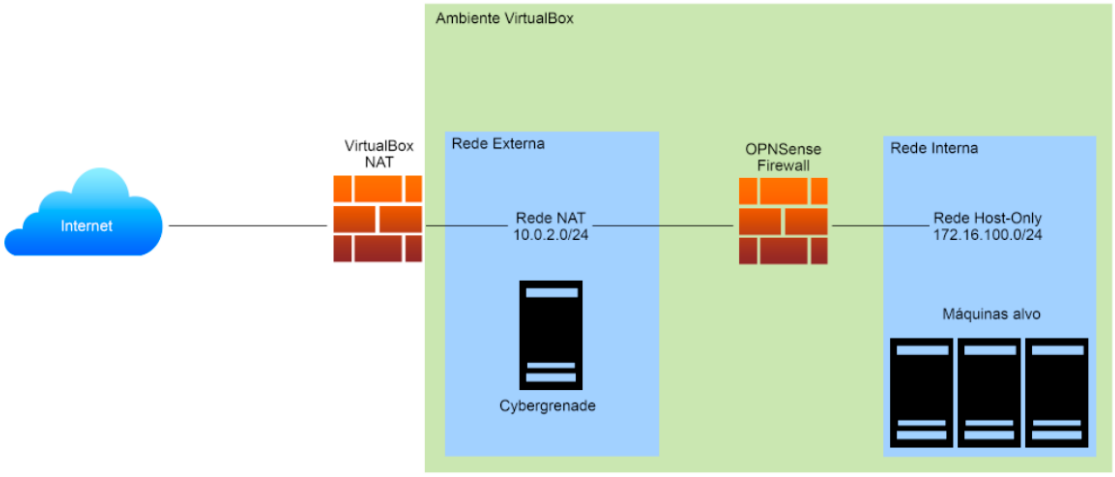

Figura 3. Diagrama do ambiente de teste do laboratório. [Fonte: O Autor]

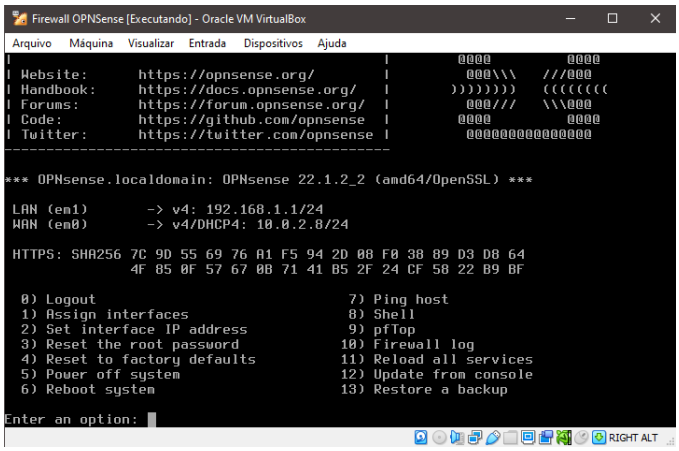

Figura 4. Tela da máquina virtual do *OPNSense.* [Fonte: O Autor]

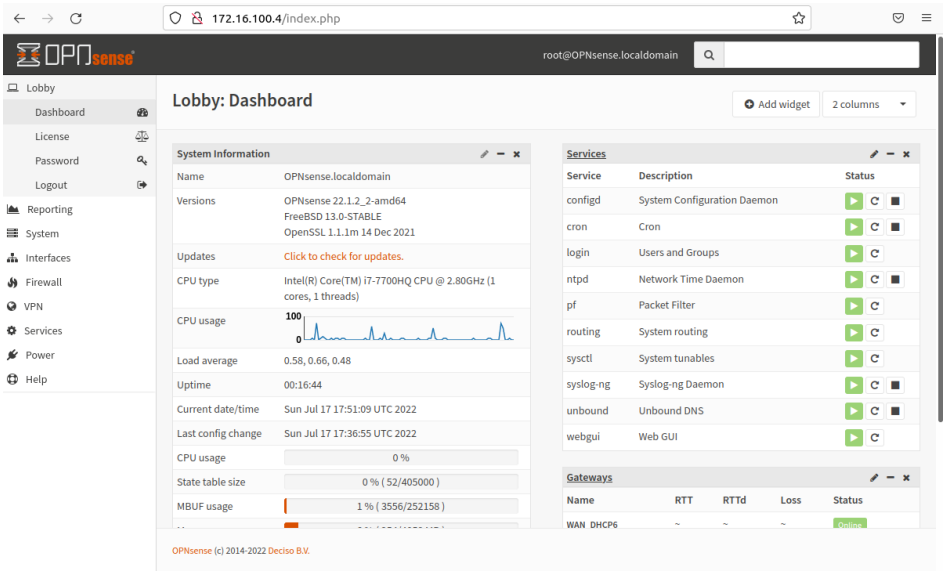

Figura 5. Interface Gráfica do painel de controle do OPNSense [Fonte: O Autor]

Para execução dos testes, foi executado o *framework* na rede alvo para verificação da alcançabilidade, primeiramente com o *Cybergrenade* rodando diretamente no alvo sem passar pelo *firewall*. Após obter os resultados e relatórios de *scan* do *OpenVAS*, e possivelmente sucesso na invasão e uma conexão reversa feita com o atacante, será feita a execução do *framework* com verificação dos pacotes que passam pelo *firewall*. A primeira máquina alvo escolhida foi o *Metasploitable 2*, uma máquina virtual fornecida pela *Rapid7* utilizada para a prática de técnicas de invasão que contém diversas vulnerabilidades, além de já possuir um servidor *web* funcional, facilitando os testes direcionados.

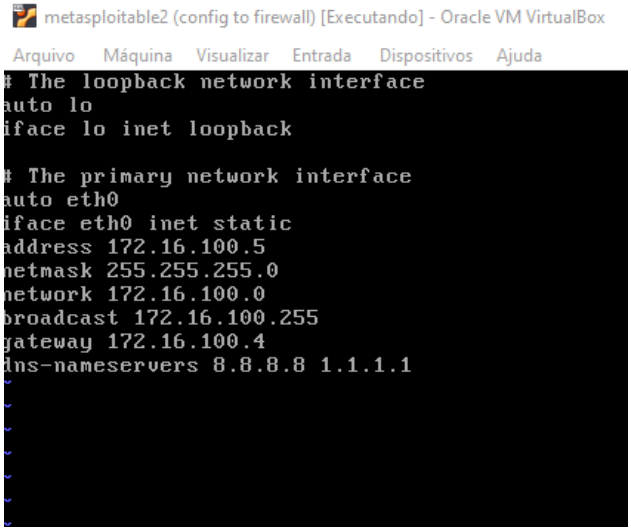

Figura 6. Configurações para funcionamento do Metasploitable com o Firewall [Fonte: O Autor]

Uma segunda máquina-alvo também foi escolhida, um Windows Server 2012 com a versão vulnerável do *Samba* instalada, a qual possui a falha enumerada *CVE-2017-0143* ou *MS17-010*, uma falha que permitia ao atacante executar código arbitrário por meio de uma falha encontrada no serviço *smbd*, também chamada de *EternalBlue*.

5. Funcionamento do Modelo:

No primeiro momento, ao testar o algoritmo sem a presença do *firewall*, o retorno que se obteve no relatório gerado pelo *OpenVAS* foi de de vinte e oito *CVEs [Common Vulnerability and Exposures]* na máquina *Metasploitable 2*, indicando que nesse sistema existem falhas registradas e catalogadas, normalmente utilizados estes identificadores para se justificar correções e *patches* em empresas que usam determinado *software*, além de passar o conhecimento para os que usam uma versão defasada do programa e entenderem os riscos aos quais estão expostos caso não busquem uma atualização.

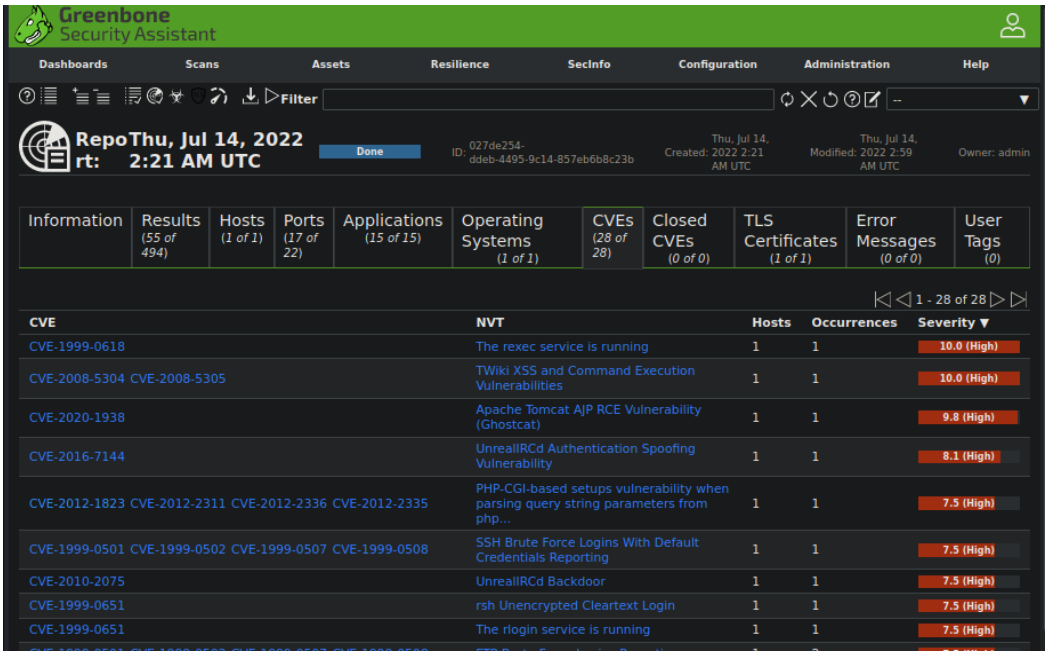

Figura 7. Relatório do *OpenVAS* [Fonte: O Autor]

Além disso, o *framework* conseguiu obter a conexão reversa com a máquina alvo por meio do *CVE-2012-1823* e pelo *CVE-2007-2447*. O primeiro destes, é uma falha presente em uma falta de validação na requisição feita ao servidor *PHP* que permite que atacantes construam uma chamada mal-intencionada e executem código não autorizado diretamente no sistema [MITRE, 2012]. O segundo, abusa da funcionalidade MS-RPC de chamada de procedimento remota no sistema de compartilhamento de arquivos *smbd* no *Samba 3.0.0*, que possibilita ao atacante enviar meta caracteres [normalmente utilizados para referenciar outras funções além de seu sentido literal] e rodar código arbitrário no sistema.

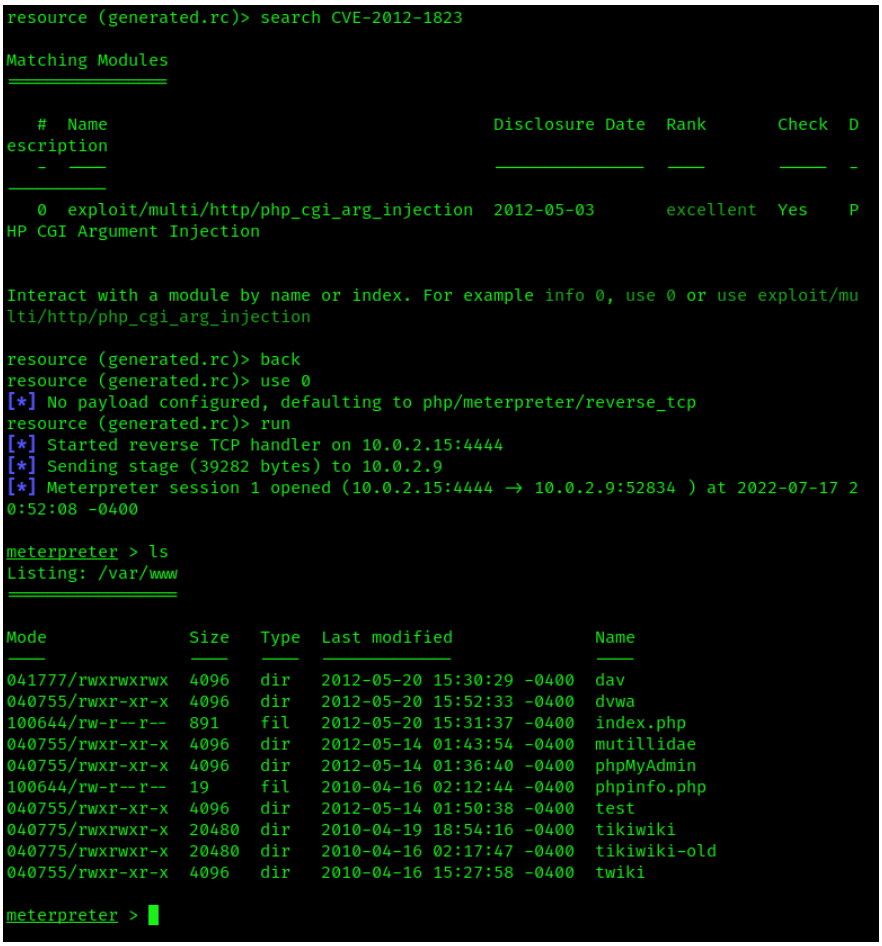

Figura 8. Conexão Reversa pelo exploit do CVE-2012-1823 [Fonte: O Autor]

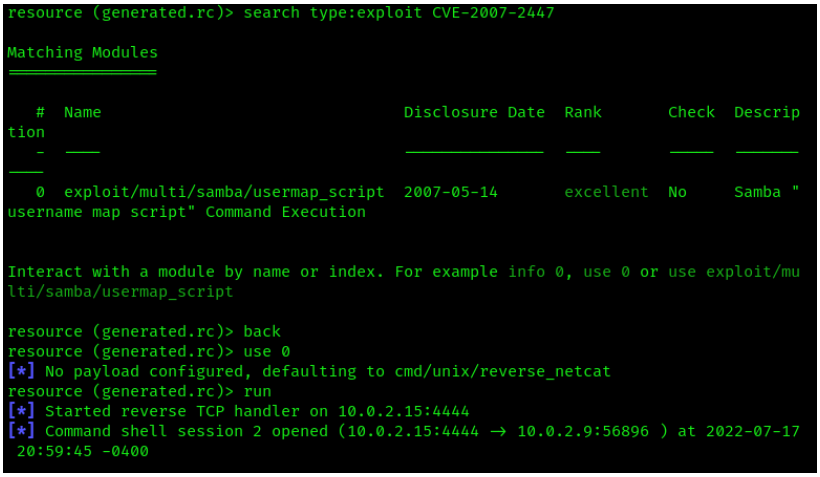

Figura 9. Conexão Reversa pelo exploit do CVE-2007-2447 [Fonte: O Autor]

Nas duas situações, o *Metasploit Framework* possuía o *exploit* [responsável por se aproveitar da falha propriamente dita] dessas vulnerabilidades e nos permitiu acesso administrador ao alvo via linha de comando. Da mesma forma, o *Cybergrenade* conseguiu se aproveitar da falha encontrada no Windows Server 2012 sem dificuldades, visto que o *Metasploit* também possui os recursos para a invasão se aproveitando da falha do *MS17-010*, obtendo uma *shell* reversa com o servidor.

Ao ativar o *firewall* em seu modo mais básico, com um simples redirecionamento de portas para que o serviço fique em uma rede interna e o atacante tenha de passar pelo *gateway* do *OPNSense*, é visto que existe uma queda no número de portas escaneadas visto que agora a disponibilização dos serviços expostos é diretamente controlada pelo mecanismo de proteção. Além disso, ao se iniciar o escaneamento de vulnerabilidades automatizado pelo *OpenVAS*, não foi detectado nenhuma falha devido a regra de bloqueio de requisições vindas desse software presentes no *firewall*. Na máquina do Windows Server 2012, também foi visto a detecção da porta 139 rodando o serviço do *Samba*, porém ao se iniciar os testes com a ferramenta de escaneamento, o *OPNSense* bloqueou os pacotes de rede que continham dados indicando ter como fonte uma ferramenta de *pentest*.

Interface	Time	Source	Destination	Proto	Label
wan	2022-07-19T00:42:33	10.0.2.15	10.0.2.8	icmp	Default deny rule
wan	2022-07-19T00:37:22	10.0.2.15:461	10.0.2.8:38877	tcp	Default deny rule
wan	2022-07-19T00:37:22	10.0.2.15:460	10.0.2.8:38877	tcp	Default deny rule
lan	2022-07-19T00:37:22	172.16.100.5:80	10.0.2.15:459	tcp	Default deny rule
wan	2022-07-19T00:37:22	10.0.2.15:443	10.0.2.8:42980	udp	Default deny rule
wan	2022-07-19T00:37:22	10.0.2.15	10.0.2.8	icmp	Default deny rule
wan	2022-07-19T00:37:22	10.0.2.15	10.0.2.8	icmp	Default deny rule
wan	2022-07-19T00:37:22	10.0.2.15:461	10.0.2.8:38877	tcp	Default deny rule
wan	2022-07-19T00:37:22	10.0.2.15:460	10.0.2.8:38877	tcp	Default deny rule
lan	2022-07-19T00:37:22	172.16.100.5:80	10.0.2.15:459	tcp	Default deny rule
wan	2022-07-19T00:37:21	10.0.2.15:443	10.0.2.8:42980	udp	Default deny rule

Figura 10. Bloqueio de pacotes do *firewall* [Fonte: O Autor]

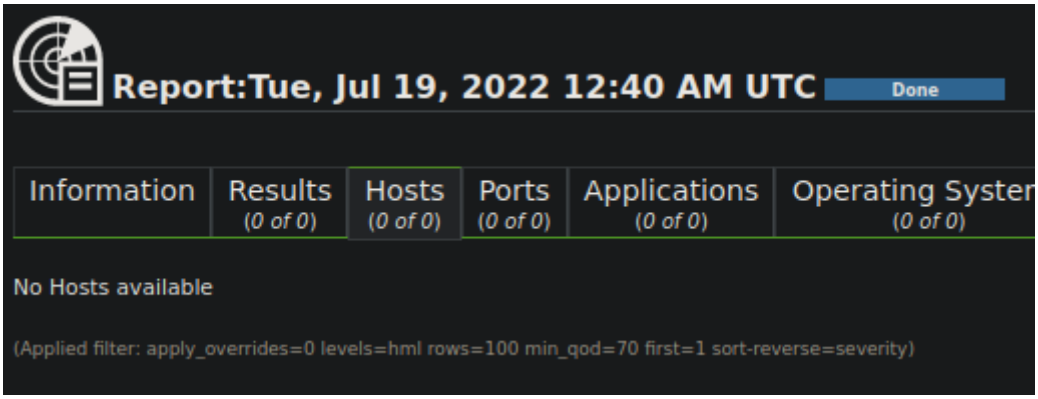

Figura 11. Report com nenhuma porta encontrada [Fonte: O Autor]

6. Discussão:

Ao se analisar os resultados do teste, é possível perceber que o algoritmo por si só é muito simples, e que se puramente executado, depende da falta de um sistema de segurança presente na rede. Com isso, deve-se fazer uma análise dos parâmetros que se pode alterar no código para que, com a experiência do profissional, se façam ajustes no código para uma execução mais incisiva e que burle os mecanismos de proteção existentes na rede.

Da mesma forma, é fato que a simples existência de um *firewall* na rede impediu a execução de um código que poderia ser de uma fonte não autorizada. A linha de

pensamento que se aplica na metodologia de Segurança em Profundidade prega a existência desses dificultadores em todas as camadas de um sistema, tornando mais difícil para um invasor causar maiores danos mesmo tendo adentrado no programa.

Mesmo que as máquinas escolhidas para execução dos testes não fossem desprovidas de falhas conhecidas, é importante relembrar que o framework possui um limitador em decorrência da necessidade de existência de vulnerabilidades catalogadas, reforçando a ideia de que para a execução bem sucedida de um pentest, deva-se agregar tanto a agilidade com automatização de ferramentas quanto o conhecimento técnico do profissional para visão crítica e pensamentos "fora da caixa" que um teste de intrusão bem sucedido requer, abrindo a possibilidade de exploração de exploits desconhecidos.

É possível concluir que o trabalho concluiu seu objetivo, revelando a eficácia do framework em um ambiente mais próximo de um empresarial, que no caso foi quebrada pela presença do firewall, com a invasão das máquinas virtuais impedidas por meio do bloqueio de pacotes. A importância dos testes também se mostrou na evidência de mostrar que um

7. Trabalhos Futuros:

Com a oportunidade de modificação do Cybergrenade para maior maleabilidade durante a execução, é aberta a possibilidade de tornar possível transpassar barreiras lógicas como detectores de intrusão automatizados, mantendo a facilidade de implementação e leveza de processamento. É também viável a adaptação do framework para maior utilização de poder computacional, visto que originalmente se foi pensado em economizar neste quesito, escalando para uso em sistemas robustos e aumentando o leque de possibilidades e poder de invasão.

8. Referências:

Akkiraju, A., Gabay, D., Yesilyurt, H. B., Aksu, H. e Uluagac, S. [2017]. Cybergrenade: Automated exploitation of local network machines via single board computers. In 2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems [MASS]. . IEEE.

Chowdhary, A., Huang, D., Mahendran, J. S., et al. [2020]. Autonomous security analysis and penetration testing. In 2020 16th International Conference on Mobility, Sensing and Networking [MSN]. . IEEE.

Cibersegurança: investimentos vão superar R$ 5 bilhões no país em 2022 [[S.d.]]. https://www.convergenciadigital.com.br/Seguranca/Ciberseguranca%3A-investimentos-vao-superar-R%24-5-bilhoes-no-pais-em-2022-59391.html?UserActiveTemplate=mobile, [accessed on Jul. 18].

Cort, N. D. [24 jun 2021]. 51% do total das operações bancárias foram feitas pelo celular durante pandemia. https://investnews.com.br/financas/51-do-total-das-operacoes-bancarias-foram-feitas-pelo-celular-durante-pandemia/, [accessed on Jul. 18].

Crimes digitais crescem pós-pandemia e provocam corrida por ciberseguros [27 jun 2022]. https://valor.globo.com/patrocinado/dino/noticia/2022/06/27/crimes-digitais-crescem-pos-pandemia-e-provocam-corrida-por-ciberseguros.ghtml, [accessed on Jul. 18].

CVE-Website [[S.d.]]. https://www.cve.org/About/Overview, [accessed on Jul. 18].

Duarte, S. [2002]. Arquitecturas do Firewall. http://www.ipg.pt/user/~sduarte/rc/trabalhos/Firewalls/Arquitecturas/arquitecturas_do_firewall.htm, [accessed on Jul. 31].

Features [[S.d.]]. https://www.kali.org/features/, [accessed on Jul. 18].

Firewall: o que é, como funciona e qual a importância para empresas? [25 jul 2022]. https://www.microserviceit.com.br/firewall/, [accessed on Jul. 31].

ISECOM [[S.d.]]. https://www.isecom.org/, [accessed on Jul. 18].

Janone, L. [26 maio 2022]. Procura por seguros cibernéticos cresce mais de 40% no 1o trimestre, diz pesquisa. https://www.cnnbrasil.com.br/business/procura-por-seguros-ciberneticos-cresce-mais-de-40-no-1o-trimestre-diz-pesquisa/, [accessed on Jul. 18].

Kadam, S. P., Mahajan, B., Patanwala, M., Sanas, P. e Vidyarthi, S. [2016]. Automated Wi-Fi penetration testing. In 2016 International Conference on Electrical, Electronics, and Optimization Techniques [ICEEOT]. . IEEE.

Li, Z. [2016]. Typical configuration of enterprise networks. https://www.researchgate.net/figure/Typical-configuration-of-enterprise-networks_fig5_313456440, [accessed on Jul. 31].

Macedo, I. [3 abr 2019]. Segurança Ofensiva/Defensiva. O que são? https://www.linkedin.com/pulse/seguran%C3%A7a-ofensivadefensiva-o-que-s%C3%A3o-iran-m/?originalSubdomain=pt, [accessed on Jul. 18].

Metasploitable 2 [[S.d.]]. https://docs.rapid7.com/metasploit/metasploitable-2/, [accessed on Jul. 18].

NIST NVD analysis - record vulnerabilities in 2021 [8 dez 2021]. https://www.redscan.com/news/nist-nvd-analysis-2021-record-vulnerabilities/, [accessed on Ago. 14].

O que é cibersegurança? [10 jan 2022]. https://www.kaspersky.com.br/resource-center/definitions/what-is-cyber-security, [accessed on Jul. 18].

OpenBSD [2011]. https://www.openbsd.org/, [accessed on Ago. 2].

Penetration testing [26 set 2019]. https://www.offensive-security.com/penetration-testing/, [accessed on Jul. 18].

Stefinko, Y., Piskozub, A. e Banakh, R. [2016]. Manual and automated penetration testing. Benefits and drawbacks. Modern tendency. In 2016 13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science [TCSET]. . IEEE.

The penetration testing execution standard [[S.d.]]. http://www.pentest-standard.org/index.php/Main_Page, [accessed on Jul. 18].

WSTG - Latest [[S.d.]]. https://owasp.org/www-project-web-security-testing-guide/latest/3-The_OWASP_Testing_Framework/1-Penetration_Testing_Methodologies, [accessed on Jul. 18].

What Is an Exploit? [9 jul 2018]. . https://www.cisco.com/c/en/us/products/security/advanced-malware-protection/what-is-exploit.html, [accessed on Ago. 1].

Zhou, S., Liu, J., Hou, D., Zhong, X. e Zhang, Y. [2021]. Autonomous penetration testing based on improved deep Q-network. Applied sciences [Basel, Switzerland], v. 11, n. 19, p. 8823.

Zhou, T.-Y., Zang, Y.-C., Zhu, J.-H. e Wang, Q.-X. [2019]. NIG-AP: a new method for automated penetration testing. Frontiers of Information Technology & Electronic Engineering, v. 20, n. 9, p. 1277–1288.

Software Testing Help [[S.d.]]. https://www.softwaretestinghelp.com/ethical-hacking-tools/, [accessed on Jul. 18].