V Encontro de Iniciação Científica e Tecnológica
V EnICT
ISSN: 2526-6772
IFSP – Câmpus Araraquara
22 e 23 de outubro de 2020

AUTOMATIZAÇÃO DE UMA FERRAMENTA PARA REALIZAÇÃO DE PENTEST UTILIZANDO SHELL SCRIPT

RAFAEL LEHNENN OSÓRIO[1], LUCAS DE ARAUJO OLIVEIRA[2]

[1] Graduando em Análise e Desenvolvimento de Sistemas, IFSP Campus Barretos, r.osorio@aluno.ifsp.edu.br
[2] Coautor docente em Análise e Desenvolvimento de Sistemas, IFSP Campus Barretos, lucas@ifsp.edu.br
Área de conhecimento (Tabela CNPq): Sistemas de informação - 1.03.03.04-9

RESUMO: Conectando-se dois ou mais computadores de forma que consigam trocar informações entre eles, criamos uma rede de computadores. Tratando-se da rede de internet, são incontáveis dispositivos conectados, mas como podemos garantir que existe segurança neste meio? Assim destacam-se os cibercriminosos, indivíduos com conhecimentos da segurança de computadores, os quais buscam explorar falhas afim de tomar vantagem de alguma informação de uma vítima. Atualmente, o maior ativo do mercado é a informação, com o objetivo de minimizar os riscos de grandes empresas sofrerem ataques que podem prejudicá-las, novas metodologias foram surgindo, tais como o Pentest. Baseia-se em uma série de passos com finalidade de detectar e explorar vulnerabilidades existentes nos sistemas, ou seja, simular um ataque que um cibercriminoso efetuaria, afinal as ferramentas utilizadas por ambos os lados são as mesmas. Usufruindo da metodologia pentest, pensou-se em desenvolver uma ferramenta o mais autônoma possível a qual executa um escaneamento da rede de computadores, buscando portas abertas, serviços e caso certa vulnerabilidade específica esteja presente, a partir de algumas ferramentas já existentes ela é executada. Por fim gera-se um relatório automático simples sobre a falha, exibindo o risco que tal vulnerabilidade pode apresentar caso explorada.

PALAVRAS-CHAVE: Kali Linux; Redes; Segurança; Shell Script.

INTRODUÇÃO

Quaisquer sistemas os quais utilizam a rede de internet como meio de comunicação possuem uma estrutura muito dinâmica e poderosa rodando sobre ela. A segurança destes sistemas acabou tornando-se um item de preocupação dos analistas e gestores em virtude de possíveis ataques virtuais. Segundo Matthew Walker existem três pilares que constituem a segurança da informação, sendo eles a confidencialidade, integridade e disponibilidade. Por mais elaborada que seja a segurança de um sistema, não podemos garantir que de fato ele esteja completamente seguro e confiável, entretanto há maneiras de minimizar os riscos. Uma das formas é o chamado Pentest (Penetration test) conhecido como teste de invasão, com o propósito fundamental de avaliar qualquer tipo de consequência que uma vulnerabilidade possa ter, de forma a elaborar um relatório de falhas para posterior correção. Segundo Jason Firch existem três modelos padrões de Pentest, White, Black e Grey Box.

O White Box é desenvolvido com conhecimento do departamento de TI. O pentester receber informações sobre a infraestrutura da rede e suas informações. Este modelo Busca simular um ataque de um membro de uma empresa o qual tenha muito conhecimento do sistema. Por outro lado, o Black Box simula um ataque de um hacker o qual não possui conhecimentos prévio sobre o sistema alvo, obrigando o pentester a buscar informações. Já o Grey Box é a mistura dos outros modelos, o pentester recebe algumas informações, mas ainda terá de fazer uma busca mais detalhada sobre o alvo até atingir seu objetivo final.

Analisando a área de segurança da informação, existem três pilares principais que a constituem, qualquer profissional deve se atentar em mantê-los da melhor característica possível para garantir que os sistemas e aplicações tornem-se o mais seguro, sendo eles citados abaixo:

Confidencialidade: A informação só deve ser acessada por entidades autorizadas. A perda deste pilar ocorre quando há quebra de sigilo de um dado, como por exemplo a senha de um usuário.

Integridade: A informação deve conter sua característica original, sem modificações. A perda deste pilar resulta em um usuário alterar os dados sem a permissão do proprietário.

Disponibilidade: A informação deve estar disponível para acesso quando necessário. A perda deste pilar resulta na informação não estar acessível, por exemplo um servidor offline.

FUNDAMENTAÇÃO TEÓRICA

Nos primórdios das redes de computadores, cada fabricante desejava desenvolver sua própria arquitetura que permitisse a comunicação dos dispositivos. A ISO (International Organization for Standardization) órgão de padronização, criou o modelo OSI (Open System Interconnection) o qual foi mantido como modelo de referência. O modelo baseia-se em uma arquitetura dividida em sete camadas. Cada protocolo realiza inserção de uma funcionalidade de acordo com cada camada específica. Hierarquicamente temos as camadas de aplicação, apresentação, sessão, transporte, rede, enlace de dados e física (CANALTECH, 2016).

Segundo Tanenbaum o Modelo OSI não é uma arquitetura de redes, pois não especifica os serviços e protocolos exatos que devem ser usados em cada camada. Ele apenas informa o que cada camada deve fazer. Assim mostrando outro modelo de rede, conhecido como TCP/IP. Paralelamente ao desenvolvimento do modelo OSI, a Agência de Projetos de Pesquisas Avançada (ARPA) do Departamento de Defesa dos Estados Unidos (DoD) deu início ao desenvolvimento de um protocolo para comunicação de dados que posteriormente foi batizado como TCP/IP (Transmission Control Protocol/Internet Protocol). O modelo OSI foi utilizado como orientação de estudos enquanto o modelo TCP/IP era realmente utilizado. Esse modelo conta com apenas quatro camadas conhecidas hierarquicamente como aplicação, transporte, internet e rede (REIS, 2017).

O TCP/IP recebeu este nome levando-se em consideração os duas principais protocolos do modelo OSI: TCP e IP. O protocolo TCP pertence a camada 4 e sua principal funcionalidade é garantir a entrega dos dados enviados, além de controlar o fluxo de transmissão e abrir e encerrar conexões. Semelhante ao protocolo IP possui um cabeçalho de 32 bits de largura. O protocolo IP pertence a camada três e é responsável por endereçar e encaminhar os pacotes que trafegam pela rede mundial de computadores.

Existem várias metodologias para pentests e uma delas é proposta por Matthew Walker, dividida em cinco fases conforme a seguir:

Reconhecimento: Buscar obter informações sobre o alvo específico e sua rede. Coletar informações como o sistema operacional, plataforma de execução, versões de servidor web e etc. Esta fase conta com a utilização do footprinting ativo, quando interage diretamente com o alvo e o passivo, quando não há interação. O atacante pode buscar por ameaças zero-day, ou seja, vulnerabilidades de softwares de computador desconhecidas ou não tratadas por aqueles que deveriam atenuar.

Escaneamento: Executar o escaneamento da rede, em grande parte feita com o Nmap, o qual consegue encontrar portas abertas e serviços rodando no sistema. O modo privilegiado tem acesso a todas opções possíveis, o Nmap[1] fornece muito mais informações do que normalmente estariam disponíveis para um usuário normal. Dependendo dos filtros no comando, o processo pode tornar-se "silencioso" ou "barulhento" dentro da rede.

Obtenção de Acesso: O atacante ganha acesso, podendo ser a nível do sistema operacional, aplicação ou rede. Pode-se elevar os privilégios para obter controle completo do sistema. Nesta fase destacam-se ferramentas de quebra de senha para obter credenciais de usuários. Outra importante ferramenta muito utilizada nesta fase é o Metasploit[2], que objetiva explorar diversas vulnerabilidades.

Manter o Acesso: Procura reter o controle sobre o sistema, mantendo sua exclusividade com backdoors e trojans. Podendo alterar os dados, aplicações e configurações do sistema além de possibilitar o uso do sistema como uma máquina de ataque contra outras máquinas na rede (MACEDO, 2016). Uma ferramenta comum no ambiente de Pen Test e capaz de efetuar um backdoor é o netcat[3], um utilitário capaz de ler e gravar dados de conexões de rede utilizando protocolos TCP/UDP. Assim criando uma forma de se conectar com a máquina alvo mais eficiente e rápida.

Apagar os Rastros: A última fase tem como objetivo esconder qualquer ato malicioso, mantendo acesso ao sistema sem ser percebido. Pode-se efetuar a exclusão ou modificação dos logs (registro de eventos) do servidor e das aplicações (MACEDO, 2016).

Para cada uma das fases, existem diversas técnicas e ferramentas específicas, porém elas não estão interconectadas. O objetivo do software, é desenvolver uma ferramenta automática que simule determinados pontos dessas fases, a partir de um endereço de IP, efetuará automaticamente um escaneamento da rede, procurando por hosts ativos, portas abertas, serviços e o endereço MAC do dispositivo, e posteriormente executará ferramentas externas para identificar falhas no alvo. Existem diversas vulnerabilidades divulgadas, as quais podem ser encontradas facilmente na internet. No caso foi usada uma vulnerabilidade muito comum aos sistemas operacionais Windows e uma ao Linux. Caso possua tal falha, executará um determinado pedaço do código e ganhará acesso remoto ao dispositivo, desta forma permitindo que o usuário possa criar alguma maneira de manter o acesso, ou simplesmente terminar a execução e perder o controle da máquina alvo.

METODOLOGIA

Visando o desenvolvimento do software automático, encontram-se alguns pontos chaves para maior entendimento do processo. Um breve estudo sobre redes de computadores, procurando entender os dois principais modelos de rede, sendo eles o OSI e o TCP/IP, analisando seus protocolos e comportamentos. Além de uma introdução à segurança da informação, entendendo os principais pilares da área e como são encontradas as vulnerabilidades nos sistemas.

Um ambiente de teste deve ser construído, realizar ataques em redes não autorizadas resultam a punições devido leis específicas sobre a área. O VirtualBox[4] é um software comumente utilizado nestes casos pois permite a criação de máquinas virtuais dentro da máquina física, ou seja, o computador consegue criar outras máquinas as quais rodam normalmente. Tais máquinas virtuais devem especificar o sistema operacional que irá rodar. O mais utilizado neste cenário é o Kali Linux[5], projetado especificamente para profissionais da área da segurança por possuir diversos programas que auxiliam na análise e exploração. Necessita-se de outras máquinas virtuais para que sirvam de máquinas alvo do teste de ataque, conectando-as à mesma rede e garantido um ambiente de teste controlado. Máquinas virtuais com o sistema operacional Windows e Linux garantem a execução do software, devido vulnerabilidades existentes em ambos os sistemas.

Montado o ambiente de testes, analisa-se o que será usado do sistema operacional para desenvolver o software. Tratando-se de um software que busca automatizar tarefas, uma linguagem de programação destaca-se por sua facilidade de criar scripts, o Shell Script[6]. A linguagem permite desenvolver de forma fácil e prática as operações necessárias para o funcionamento do software.

Concluído as operações, deu-se início ao desenvolvimento do software, explorando funcionalidades da linguagem de programação e reunindo com recursos e ferramentas do Kali Linux. Assim procurando o máximo possível automatizar o processo, fazendo com que o usuário digite apenas o endereço de IP do alvo, e a partir dele, executa uma série de comandos. Deste modo, o software tornou-se capaz de efetuar um ataque inteiro automatizado, conseguindo efetuar análises de vulnerabilidades e executar código malicioso na máquina alvo caso possua os sistemas operacionais usados como alvos, Windows e Linux.

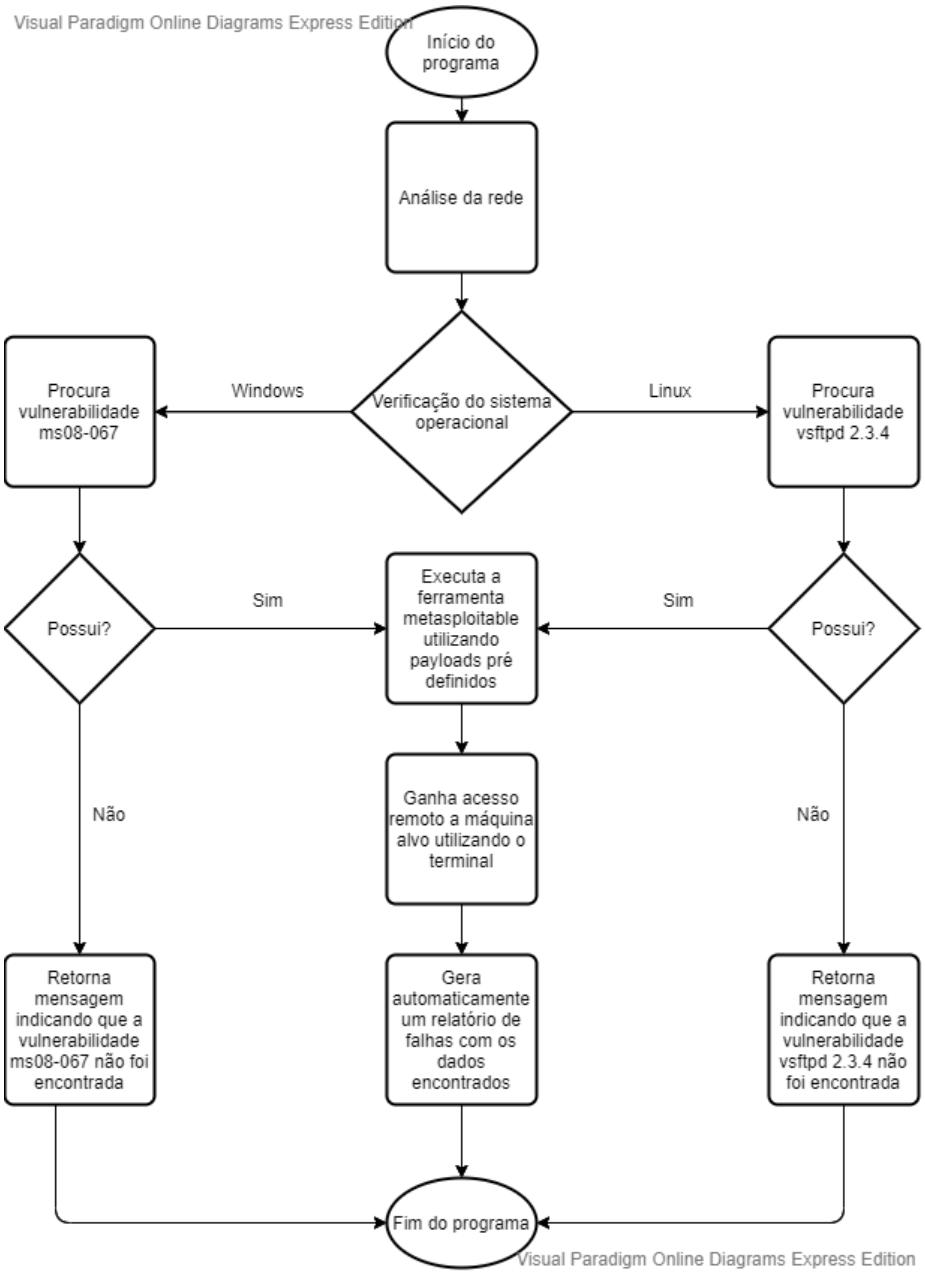

FIGURA 1. Fluxograma de controle.
Fonte: Elaborada pelo autor (2020)

RESULTADOS E DISCUSSÃO

O desenvolvimento da ferramenta automática para pentest foi concluída utilizando as principais ferramentas para análise e ataques cibernéticos. Foi necessário um levantamento bibliográfico sobre as fases e um estudo detalhado de cada uma das técnicas e ferramentas utilizadas em cada passo. A ferramenta se torna viável devido as expressões regulares buscando vulnerabilidades e retornando-as para que o analista tenha em mãos tais informações que poderiam passar despercebidas. As figuras 2 e 3 ilustram respectivamente o script em execução e o ganho de acesso a máquina alvo.

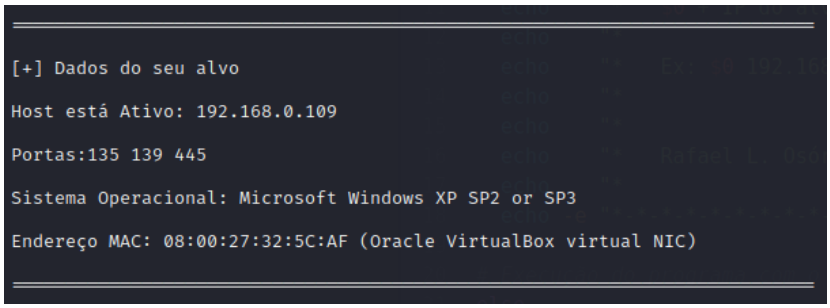

FIGURA 2. Execução do software identificando dados do alvo
Fonte: Elaborada pelo autor (2020)

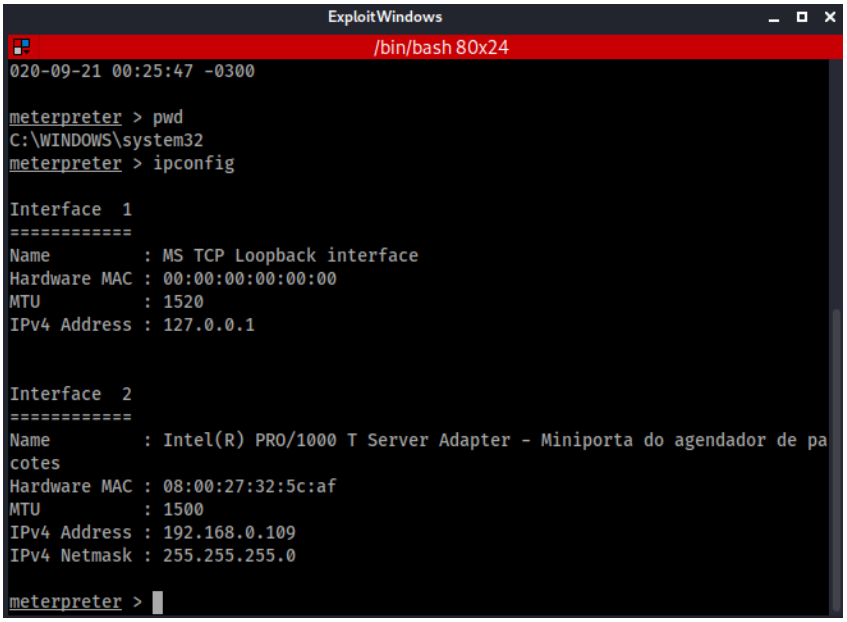

FIGURA 3. Ganho de acesso a máquina alvo
Fonte: Elaborada pelo autor (2020)

CONCLUSÕES

De fato, inúmeras aplicações possuem diversas vulnerabilidades, tornando possível que um conhecedor do assunto, possa invadir e prejudicar um sistema, seja por parte de profissionais os quais buscam justamente encontrar tais falhas para posterior correção, quanto de um mal-intencionado que apenas faz por diversão ou para ganhar vantagem da situação. Explorando de um sistema operacional livre e seus recursos, observou-se possível desenvolver uma nova ferramenta a qual seria capaz de fazer uma análise simples de rede, explorando se a máquina alvo possui determinada vulnerabilidade, caso possuir, através de códigos maliciosos e softwares disponíveis, permite que o atacante consiga explorar a falha. A partir de duas vulnerabilidades já divulgadas, uma para Windows e outra para Linux, as quais garantem acesso remoto a máquina alvo, permitiu-se o desenvolvimento de um código capaz de explorar de forma automática tais falhas que estão presentes em diversos computadores da rede de internet. Desta forma, provou-se possível um atacante adquirir controle sobre uma máquina alvo de forma remota, podendo-se automatizar o processo através de scripts.

REFERÊNCIAS

[1] **Nmap** disponível em: https://nmap.org/

[2] **Metasploit** disponível em: https://www.metasploit.com/
[3] **Netcat** disponível em: https://pt.wikipedia.org/wiki/Netcat

[4] **VirtualBox** disponível em: https://www.virtualbox.org/

[5] **Kali Linux** disponível em: https://www.kali.org/

[6] **Shell Script** disponível em: https://pt.wikipedia.org/wiki/Shell_script

BENETTI, T. **Segurança da Informação - Confidencialidade, integridade e Disponibilidade (CID)**. 20 de julho 2020. Disponível em: <https://www.profissionaisti.com.br/seguranca-da-informacao-confidencialidade-integridade-e-disponibilidade-cid>.

FIRCH, J. **What Are The Different Types Of Penetration Testing?** Disponível em: <https://purplesec.us/types-penetration-testing/>.

FRAGA, B. **Técnicas de invasão**. 2019, p.11-17.

LOSSIO, R. **Rede de computadores: Comparação entre o Modelo de Referência OSI e o TCP/IP**. 30 de outubro de 2018. Disponível em: <https://oraculoti.com.br/2018/10/30/rede-de-computadores-comparacao-entre-o-modelo-de-referencia-osi-e-o-tcp-ip/>.

MACEDO, D. **Fases de um pen test**. 4 de outubro 2016. Disponível em: <https://www.diegomacedo.com.br/fases-de-um-pentest/>.

MAYA, A. **O que são redes de computadores?** 14 de março 2020. Disponível em: <https://alcidesmaya.edu.br/blog/182-o-que-sao-redes-de-computadores>.

O que é modelo OSI? [2016?]. Disponível em: <https://canaltech.com.br/produtos/o-que-e-modelo-osi/>

REIS, R. Modelo TCP/IP - **Definição, camadas e funcionamento.** Disponível em: <http://infotecnews.com.br/modelo-tcpip/>.

SILVA, H. S. **Quais são os tipos de PenTest?**. Disponível em: <https://suporte.siteblindado.com.br/hc/pt-br/articles/115002317051-Quais-s%C3%A3o-os-tipos-de-PenTest->.

VIEGAS, J. **O que é (e para que serve) o pentest**. Disponível em: <https://blog.onedaytesting.com.br/o-que-e-pentest-e-para-que-serve/>.

WALKER, M. **CEH Certified Ethical Hacker All-In-One Exam Guide, Fourth Edition** 2019.