



**UNIVERSIDADE DE ITAÚNA
PRÓ-REITORIA DE ENSINO
COORDENAÇÃO DE CIÊNCIA DA COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**DAVI VENTURA CARDOSO PERDIGÃO
EDMILSON LINO CORDEIRO
ERIC HENRIQUE DE CASTRO CHAVES**

**A UTILIZAÇÃO DE PENTEST COMO FERRAMENTA DE SEGURANÇA
OFENSIVA PARA ANÁLISE DE VULNERABILIDADES EM
ORGANIZAÇÕES**

**ITAÚNA
2023**

RESUMO:

A segurança da informação é um dos maiores desafios enfrentados pelas organizações modernas. Ataques cibernéticos estão se tornando cada vez mais sofisticados e frequentes, o que torna essencial a implementação de medidas de segurança para proteger as informações e os sistemas das organizações. Uma das formas mais eficientes de garantir a segurança é a realização de testes de segurança ofensiva, conhecidos como *Pentests*. Este artigo discute a importância dos *Pentests* como ferramenta de segurança ofensiva para análise de vulnerabilidades em organizações, apresentando sua definição, objetivos, tipos e os benefícios que podem trazer para as empresas.

Palavras-chave: Segurança da informação, ataques cibernéticos, medidas de segurança, testes de segurança ofensiva, *Pentests*, análise de vulnerabilidades, organizações.

SUMÁRIO:

1. INTRODUÇÃO.....	4
1.1 - CONTEXTUALIZAÇÃO.....	4
1.2 - JUSTIFICATIVA.....	4
1.3 - OBJETIVOS.....	5
1.4 - HIPÓTESES.....	6
2. CONCEITOS E FUNDAMENTOS.....	6
2.1 - TIPOS DE ATAQUES.....	6
2.1.1 - Captura e análise de pacotes.....	6
2.1.2 - Falsificação de pacotes.....	7
2.1.3 - Envenenamento de cache DNS.....	7
2.1.4 - Negação de serviço.....	8
2.1.5 - Buffer Overflow.....	9
2.1.6 - Injeção de DLL.....	9
2.1.7 - Sequestro de sessão.....	10
2.1.8 - Quebra de senhas.....	10
2.1.9 - Engenharia Social.....	11
2.2 - TIPOS DE PENTEST.....	11
2.2.1 - Termo de Responsabilidade e Confidencialidade.....	12
2.2.2 - Relatório: Sumário Executivo e Laudo Técnico.....	13
2.2.3 - Preparação do Ambiente de Testes.....	14
3. REFERENCIAL TEÓRICO.....	15
3.1 - ETAPAS DO PENTEST.....	15
3.1.1 - Planejamento.....	16

3.1.2 - Coleta de informações.....	16
3.1.3 - Análise de vulnerabilidades.....	16
3.1.4 - Exploração de vulnerabilidades.....	16
3.1.5 - Pós - exploração.....	17
3.1.6 - Relatório e recomendações.....	17
3.2 - BACKTRACK.....	17
3.2.1 - OpenVAS.....	18
3.2.2 - Metasploit Framework.....	19
3.2.3 - Ferramenta de engenharia social.....	20
4. CONCLUSÃO.....	21
5. REFERÊNCIAS BIBLIOGRÁFICAS.....	21

1. INTRODUÇÃO

1.1 - CONTEXTUALIZAÇÃO

O *Pentest* é uma técnica muito utilizada por empresas de diferentes setores para avaliar a segurança de seus sistemas e infraestrutura de tecnologia da informação. Através dele, é possível identificar potenciais vulnerabilidades em sistemas e aplicativos, tanto internos quanto externos, que possam ser exploradas por atacantes mal-intencionados.

Essa técnica consiste em simular ataques cibernéticos que podem ser realizados por *hackers* para verificar a efetividade das medidas de segurança implementadas pela organização. Os testes são realizados por profissionais especializados, que utilizam ferramentas e técnicas para avaliar diferentes aspectos do ambiente de tecnologia da informação da organização, como a rede de computadores, os servidores, os aplicativos, entre outros.

Após a realização do *Pentest*, é gerado um relatório que descreve todas as vulnerabilidades encontradas e suas possíveis consequências, bem como as recomendações para corrigi-las. Essas informações permitem que a empresa possa tomar medidas proativas para aumentar a segurança, protegendo seus dados e reduzindo o risco de ataques cibernéticos.

Além disso, o *Pentest* é uma técnica importante para demonstrar a efetividade das medidas de segurança implementadas pela empresa. Através da realização periódica do mesmo, é possível verificar se as medidas de proteção estão realmente funcionando e se há alguma falha que possa ser explorada pelos atacantes. Isso aumenta a confiança dos *stakeholders* na empresa e demonstra que a empresa está tomando medidas para proteger seus dados e informações sensíveis.

1.2 - JUSTIFICATIVA

Os ataques cibernéticos são uma ameaça real e crescente para as organizações de todos os tamanhos e setores. Segundo um relatório da Verizon (2020), os ataques cibernéticos são responsáveis por mais de 40% dos incidentes de segurança em empresas de todo o mundo. As consequências desses ataques podem ser desastrosas, como perda de dados, interrupção dos negócios, perda de reputação e prejuízos financeiros. Nesse sentido, a realização de *Pentests* se apresenta como uma ferramenta fundamental para prevenir e mitigar essas ameaças, identificando as vulnerabilidades dos sistemas de informação antes que elas possam ser exploradas por *hackers*.

Além disso, a realização de *Pentests* também é importante para garantir a conformidade com normas e regulamentações de segurança da informação, como a GDPR (*General Data Protection Regulation*) na Europa e a LGPD (Lei Geral de Proteção de Dados) no Brasil. Essas normas exigem que as empresas adotem medidas de segurança adequadas para proteger os dados pessoais dos usuários, o que inclui a realização de testes de segurança periódicos.

Por isso, é fundamental que os profissionais de tecnologia da informação e segurança estejam capacitados para realizar *Pentests* de forma eficiente e eficaz, a fim de garantir a segurança e proteção dos sistemas de informação das empresas.

1.3 - OBJETIVOS

O objetivo principal deste artigo é destacar a importância dos *Pentests* como uma ferramenta de segurança ofensiva eficaz para identificar vulnerabilidades em sistemas de informação, que pode ser aplicada em organizações de todos os setores e tamanhos. Para isso, serão abordados temas como a definição do *Pentest*, seus tipos e etapas, bem como os tipos de ataques que podem ser realizados em um contexto organizacional.

Além disso, o artigo tem como objetivo discutir como os *Pentests* podem ajudar as organizações a identificar e corrigir as falhas de segurança em seus sistemas, aumentando a efetividade das medidas de proteção e reduzindo o risco de ataques cibernéticos. Também será destacada a importância da realização de *Pentests* para demonstrar a eficácia das medidas de segurança implementadas, aumentando a confiança dos *stakeholders* na organização.

Outro objetivo é apresentar os principais desafios e limitações que podem ser encontrados na realização de *Pentests* e como superá-los, para garantir que a ferramenta seja utilizada de forma eficiente e eficaz. Serão abordados temas como a seleção de ferramentas e metodologias adequadas, a gestão dos resultados obtidos e a garantia da confidencialidade das informações obtidas durante os testes.

Por fim, este artigo tem como objetivo contribuir para a disseminação do conhecimento sobre a importância dos *Pentests* como ferramenta de segurança ofensiva, promovendo uma cultura de segurança da informação mais robusta e efetiva nas organizações.

1.4 - HIPÓTESES

Com base em estudos já realizados sobre esse tema, acredita-se que o *Pentest* é uma técnica de segurança ofensiva altamente eficaz para identificar vulnerabilidades em sistemas de informação. A realização do mesmo também pode ajudar as organizações a reduzir o risco de ataques cibernéticos, fornecendo informações valiosas sobre as falhas de segurança que precisam ser corrigidas. Além disso, a adoção de medidas de segurança eficazes, incluindo a realização de *Pentests*, pode contribuir para a proteção da informação e para a confiança dos *stakeholders*, o que é fundamental para a manutenção da reputação das organizações. No entanto, é importante salientar que os resultados gerados devem ser interpretados e tratados de maneira apropriada, para garantir que as vulnerabilidades sejam corrigidas e a segurança da organização seja mantida.

2. CONCEITOS E FUNDAMENTOS

2.1 - TIPOS DE ATAQUES

Os *Pentests* podem simular diferentes tipos de ataques cibernéticos, a fim de identificar as vulnerabilidades presentes nos sistemas de uma organização. É importante que os testes sejam realizados de forma ética e legal, respeitando os limites e acordos prévios estabelecidos com a organização que está sendo testada.

A seguir, serão detalhados alguns dos principais tipos de ataques que podem ser realizados durante um *Pentest*:

2.1.1 - Captura e análise de pacotes

Na captura e análise de pacotes, o *Pentester* geralmente utiliza ferramentas como o *Wireshark* para capturar e analisar os pacotes de dados em uma rede. Essas ferramentas permitem que ele visualize o conteúdo dos pacotes e identifique possíveis informações sensíveis que estejam sendo transmitidas sem criptografia.

O objetivo desse tipo de ataque é verificar se a rede está configurada de maneira segura e se os dados estão sendo transmitidos de forma segura. O *Pentester* pode também analisar a estrutura da rede e identificar possíveis vulnerabilidades que possam ser exploradas por atacantes mal-intencionados.

A captura e análise de pacotes é uma técnica comum em *Pentests*, pois muitas vezes as informações sensíveis são transmitidas em formato de texto simples, sem nenhuma criptografia. A identificação dessas informações pode ajudar a organização a implementar medidas de segurança para proteger seus dados.

2.1.2 - Falsificação de pacotes

A falsificação de pacotes é um tipo de ataque que pode ser realizado em diferentes níveis da pilha de protocolos de rede. Isso permite que o atacante envie pacotes maliciosos que pareçam ter sido enviados por um dispositivo legítimo, enganando o destinatário e permitindo que o atacante execute uma série de atividades maliciosas. Um exemplo comum de falsificação de pacotes é o ataque de spoofing de endereço IP (*Internet Protocol*), no qual o atacante usa uma ferramenta para enviar pacotes que aparentam ter origem em um endereço IP diferente do que realmente foi usado.

Esse tipo de ataque pode ser utilizado para interceptar informações sensíveis, como senhas e dados de cartões de crédito, ou para realizar ataques de negação de serviço, que são destinados a interromper o funcionamento normal de um serviço ou aplicativo de rede.

2.1.3 - Envenenamento de cache DNS

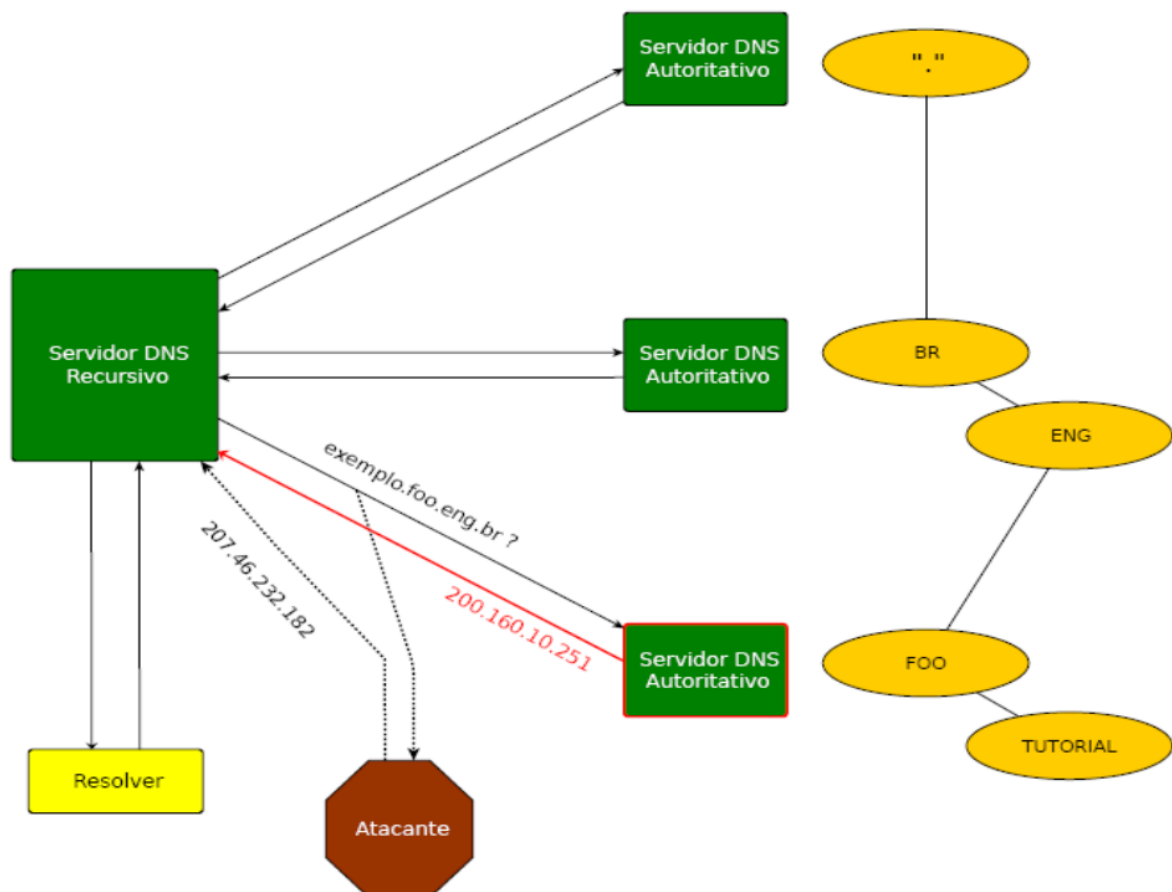
O envenenamento de cache DNS (*Domain Name System*) é uma técnica que explora a forma como os servidores DNS armazenam as informações de resolução de nomes em cache. Esses servidores mantêm um registro das consultas DNS recentes, para acelerar a resolução de nomes de domínio no futuro.

No envenenamento de cache DNS, um atacante pode enviar respostas falsas para um servidor DNS, contendo informações de resolução de nomes adulteradas. O servidor DNS armazena essas informações falsas em seu cache, permitindo que o atacante redirecione o tráfego de rede para um servidor mal-intencionado. Isso pode ser usado para interceptar informações sensíveis, como senhas e números de cartões de crédito.

Por exemplo, supondo que um usuário tente acessar o site "www.banco.com.br". O servidor DNS solicita informações de resolução de nomes ao servidor autoritativo do domínio "banco.com.br", que responde com o endereço IP correto do site. No entanto, um atacante pode interceptar essa resposta e enviar uma resposta falsa contendo um endereço IP diferente,

correspondente a um servidor mal-intencionado. O servidor DNS armazena essa resposta falsa em seu cache, permitindo que o atacante redirecione o tráfego de rede para o servidor mal-intencionado e, assim, intercepte as informações sensíveis.

Figura 1 - Esquemático de um ataque de envenenamento de cache. Fonte: De Campos e Justo, 2009.



2.1.4 - Negação de serviço (DoS)

Os ataques de negação de serviço (DoS - *Disk Operating System*) são uma ameaça comum enfrentada pelas organizações atualmente. Esse tipo de ataque pode ser conduzido de várias maneiras, mas a intenção é a mesma: tornar um serviço ou recurso inacessível para os usuários legítimos.

Um ataque DoS geralmente envolve o envio de uma grande quantidade de tráfego para um servidor ou sistema, de forma que ele não consiga mais processar as solicitações legítimas. Esse tipo de ataque pode ser conduzido por meio de técnicas como o envio de pacotes UDP (*User Datagram Protocol*) ou TCP

(protocolo de controle de transmissão), *flooding* de requisições HTTP (*HyperText Transfer Protocol*) ou HTTPS (*Hyper Text Transfer Protocol Secure*), exploração de vulnerabilidades em protocolos de rede, entre outras.

Os ataques DoS podem causar prejuízos significativos para as organizações, como a perda de dados, interrupção dos serviços prestados, queda na produtividade, entre outros. Por isso, é importante que as empresas realizem testes de segurança ofensiva, como o *Pentest*, para identificar as vulnerabilidades em seus sistemas e adotar medidas para prevenir ou mitigar esse tipo de ataque.

2.1.5 - Buffer Overflow

O *buffer overflow* é um tipo comum de vulnerabilidade em aplicativos e sistemas operacionais, especialmente em linguagens de programação como C e C++. Isso ocorre porque essas linguagens permitem que os desenvolvedores acessem diretamente a memória do sistema, o que pode levar a erros de programação que deixam o sistema vulnerável a ataques.

Basicamente, o *buffer overflow* ocorre quando um aplicativo tenta armazenar mais dados em um *buffer* do que ele pode suportar, o que leva a um estouro do *buffer* e uma sobrescrita da memória adjacente. Isso pode permitir que um atacante insira código malicioso no sistema, levando a um comprometimento do sistema e à execução de comandos maliciosos.

Para evitar o *buffer overflow*, é importante que os desenvolvedores utilizem técnicas seguras de programação, como a verificação adequada do tamanho do *buffer* e a limitação do acesso direto à memória do sistema. Além disso, a realização de *Pentests* pode ajudar a identificar vulnerabilidades de *buffer overflow* em um aplicativo ou sistema e permitir que as medidas de segurança adequadas sejam tomadas para corrigi-las.

2.1.6 - Injeção de DLL

A injeção de DLL (*Dynamic Link Library*) é um tipo de ataque que visa explorar a capacidade do *Windows* de carregar bibliotecas dinâmicas em tempo de execução. Essa técnica envolve a inserção de código malicioso em uma biblioteca dinâmica e, em seguida, a sua injeção em um processo em execução. O objetivo é executar o código malicioso no contexto do processo, permitindo que o atacante obtenha acesso não autorizado a recursos do sistema ou realize outras ações maliciosas.

Existem diferentes métodos para realizar a injeção de DLL, incluindo a injeção remota, em que o atacante explora uma vulnerabilidade para injetar a DLL em um processo remoto, e a injeção de DLL baseada em registro, em que o atacante altera o registro do *Windows* para que a DLL seja carregada automaticamente em um processo específico.

Essa é uma técnica sofisticada e pode ser difícil de detectar. Por essa razão, é importante que as organizações realizem testes de segurança ofensiva para identificar possíveis vulnerabilidades e implementem medidas de segurança para mitigar esse tipo de ataque.

2.1.7 - Sequestro de sessão

O sequestro de sessão, também conhecido como *session hijacking*, é um ataque que visa roubar informações de autenticação de um usuário que já está autenticado em um sistema ou aplicativo. Isso ocorre quando um atacante consegue capturar a sessão de um usuário legítimo e a utiliza para acessar o sistema ou aplicativo.

Existem várias formas de realizar um sequestro de sessão, mas uma das mais comuns é através do uso de um sniffer de rede, que captura os pacotes de dados que estão sendo transmitidos na rede. Quando um usuário se autentica em um sistema ou aplicativo, um *cookie* de sessão é criado e armazenado em seu computador. Esse cookie contém as informações de autenticação do usuário e é enviado em cada solicitação que o usuário faz ao servidor.

Ao capturar o *cookie* de sessão de um usuário, um atacante pode utilizá-lo para se passar pelo usuário legítimo e acessar o sistema ou aplicativo. Para evitar esse tipo de ataque, é importante que as sessões de usuários sejam adequadamente protegidas, através do uso de mecanismos como criptografia de *cookies*, limitação do tempo de validade das sessões e uso de *tokens* de autenticação de longa duração.

2.1.8 - Quebra de senhas

A quebra de senhas é um dos tipos de ataques mais comuns e utilizados pelos cibercriminosos para obter acesso a sistemas e dados protegidos por senha. A técnica de força bruta é relativamente simples, mas pode ser muito demorada, pois envolve testar várias combinações possíveis de senhas até que a correta seja encontrada.

Já a análise de *hashes* de senha envolve a obtenção do *hash* da senha armazenado em um sistema e a utilização de técnicas para descobrir a senha original a partir desse *hash*. Existem várias ferramentas disponíveis para realizar esse tipo de ataque, que podem utilizar dicionários de senhas, técnicas de combinação de palavras ou algoritmos específicos para a quebra de senhas. É importante ressaltar que, para se proteger contra esse tipo de ataque, é essencial utilizar senhas fortes, que sejam difíceis de serem adivinhadas ou descobertas através de técnicas de quebra de senhas.

2.1.9 - Engenharia Social

A engenharia social é uma técnica que explora a vulnerabilidade humana em relação à segurança da informação. Ela pode ser usada para enganar as pessoas a fornecer informações confidenciais, instalar malware em seus dispositivos, ou realizar outras ações maliciosas. Os cibercriminosos usam essa técnica porque é mais fácil convencer uma pessoa a fornecer suas credenciais de login ou instalar um *software* malicioso do que explorar vulnerabilidades técnicas em um sistema.

Durante um Pentest, a equipe de segurança pode usar técnicas de engenharia social para testar a eficácia das políticas de segurança da informação da organização. Por exemplo, eles podem enviar um e-mail de *phishing* para os funcionários da empresa e verificar quantos deles clicam no link malicioso ou fornecem informações confidenciais.

Ao realizar esse teste, a organização pode identificar funcionários que precisam de treinamento em segurança da informação ou pontos fracos em suas políticas de segurança que precisam ser corrigidos. Também é uma oportunidade para educar os funcionários sobre como identificar e evitar golpes de engenharia social, tornando a organização mais resiliente a esses ataques.

2.2 - TIPOS DE PENTEST

Existem diferentes tipos de Pentest que podem ser aplicados, mas os mais tradicionais e conhecidos são o *Black Box*, *Gray Box* e *White Box*. O *Black Box* (Caixa Preta) é o tipo mais exigente para o *Pentester*, pois ele começa do zero sem nenhuma informação da empresa contratante. Por esse motivo, ele é mais caro e mais utilizado pelas empresas que desejam simular uma invasão externa e analisar a vulnerabilidade de sua rede (BERTOGLIO; ZORZO, 2015).

O *Gray Box* (Caixa Cinza) tem como ideia mostrar uma visão intermediária entre departamentos. Nesse tipo de teste, o *Pentester* tem um breve conhecimento e limites de teste liberados pela empresa contratante, definindo os

limites dos ataques. O *Gray Box* fica entre o *Black Box* e o *White Box* (BERTOGLIO; ZORZO, 2015).

O *White Box* (Caixa Branca) é muito utilizado e facilita bastante o trabalho do *Pentester*, pois ele já tem conhecimento sobre todas as informações do sistema antes de realizar o *Pentest*. Antes de realizar um *Pentest* em uma empresa, o profissional deve conversar com os responsáveis para entender as necessidades de verificação de vulnerabilidades, tanto internas quanto externas (BERTOGLIO; ZORZO, 2015).

2.2.1 - Termo de Responsabilidade e Confidencialidade

Antes de realizar um *Pentest*, é importante formalizar um acordo de confidencialidade entre a empresa contratante e o pentester ou empresa responsável. Esse acordo, conhecido como NDA (*Non Disclosure Agreement*), estabelece os termos e limites do teste de invasão e evita que informações confidenciais sejam divulgadas ou utilizadas de forma indevida. É importante ressaltar que o NDA deve ser firmado antes do início dos testes, a fim de garantir a segurança e privacidade das informações da empresa.

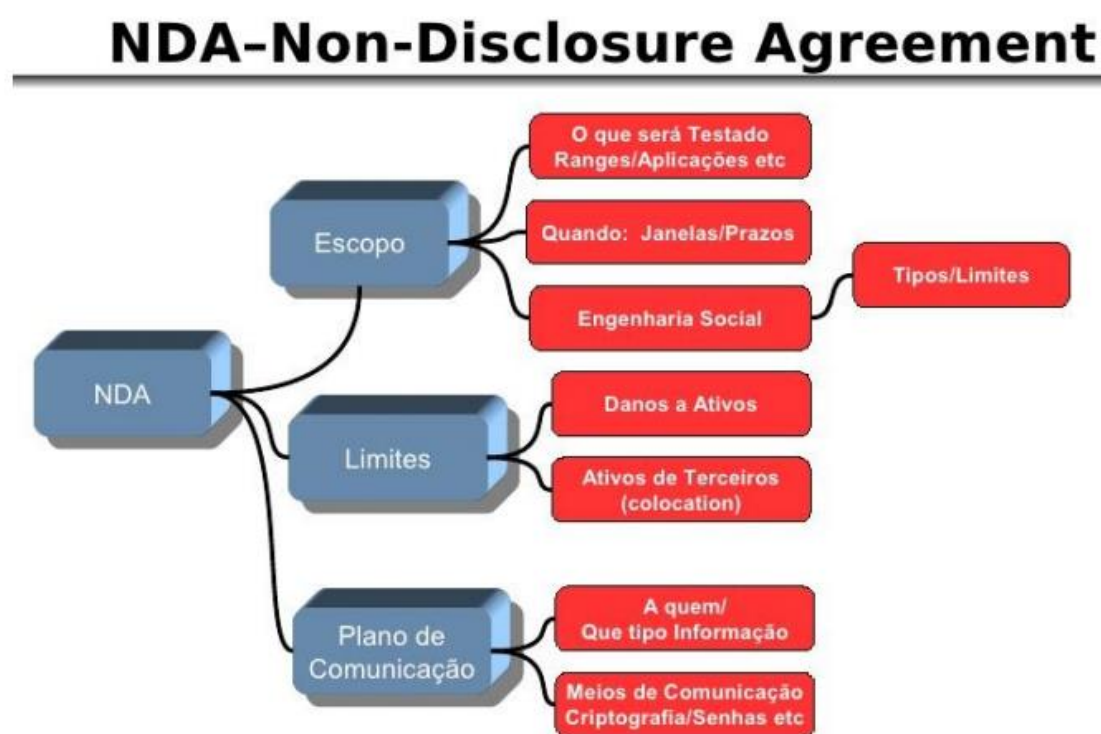


Figura 2 - Exemplo de tabela com os termos de um NDA. Fonte: ASSUNÇÃO, 2017.

O escopo do *Pentest* é um aspecto importante a ser definido no acordo de confidencialidade. Deve-se estabelecer prazos e tempo necessário para o teste,

além de considerar a aplicação de técnicas de Engenharia Social (abordada no capítulo 2.1.9), como contato com pessoas e coleta de informações sigilosas. É necessário que se estabeleçam limites claros para evitar que o *Pentester* ultrapasse o acordo com a empresa contratante (ASSUNÇÃO, 2017).

Os limites do Pentest também devem ser definidos com clareza. É importante estabelecer se a Engenharia Social será presencial ou por telefone, quais pessoas estarão envolvidas e o quão profunda será a coleta de informações. Esses limites devem ser estabelecidos para garantir que o *Pentester* não ultrapasse o acordo com a empresa contratante (ASSUNÇÃO, 2017).

O plano de comunicação é outra parte essencial do acordo de confidencialidade. Deve-se definir quem será o responsável pelo *Pentest* e a quem o *Pentester* irá se reportar. É importante que seja assinado um termo de confidencialidade, comprometendo-se a não divulgar o relatório de varreduras para nenhuma outra pessoa que não esteja envolvida no contrato. É importante destacar que as formas de pagamento não são incluídas no modelo de NDA e devem ser acordadas separadamente (ASSUNÇÃO, 2017).

2.2.2 - Relatório: Sumário Executivo e Laudo Técnico

É importante ter um entendimento completo do cliente e da empresa que contratou o serviço de *Pentest*, uma vez que o *Pentester* precisa compilar todos os problemas encontrados e entregá-los à empresa de forma detalhada. Existem dois tipos principais de relatórios de Penetration Test: o Sumário Executivo, que é um resumo para os executivos da empresa apresentando os problemas encontrados e os possíveis danos que essas falhas podem causar, bem como o tempo necessário para resolvê-los; e o Relatório Técnico, que é mais detalhado e apresenta uma descrição passo a passo das vulnerabilidades encontradas, incluindo capturas de tela e gravações de tela. É importante documentar todo o processo do *Pentest* para gerar relatórios detalhados com gráficos de análise de riscos. (ASSUNÇÃO, 2017).

Além disso, é importante lembrar que a documentação do processo de *Pentest* não se limita apenas aos relatórios finais. É essencial que o *Pentester* documente todos os passos dados durante o teste, desde a definição do escopo até a apresentação dos resultados. Essa documentação pode ser útil tanto para a empresa contratante quanto para o próprio *Pentester*, caso haja a necessidade de revisitar o processo em algum momento futuro.

Por fim, é importante que o *Pentester* entenda que o relatório de *Pentest* é um documento técnico, mas que deve ser apresentado de forma clara e objetiva para todos os envolvidos. É fundamental que a linguagem utilizada seja acessível

para o público alvo, evitando jargões técnicos que possam dificultar a compreensão dos resultados apresentados.

2.2.3 - Preparação do Ambiente de Testes

Antes de realizar testes de intrusão em um ambiente, é crucial criar um ambiente virtualizado para evitar danos ou problemas futuros para o cliente. Para demonstrar exemplos de *Pentest*, um ambiente virtualizado pode ser criado usando o sistema operacional de código aberto Kali Linux e outras máquinas virtuais para representar o alvo do cliente, como o *Windows 7 64 bits* e o *Windows Server 2003 Standard Edition*.

Antes de instalar as máquinas virtuais, é necessário obter um software de virtualização, como o *VirtualBox*, que pode ser baixado gratuitamente em seu site oficial e é de fácil instalação. O próximo passo é baixar um disco virtual (ISO) do Kali Linux, que pode ser obtido gratuitamente em seu site oficial. É recomendável usar a versão mais recente, pois oferece melhor suporte. No *VirtualBox*, a configuração do Kali Linux deve incluir pelo menos 1024 MB de memória RAM, 30GB de espaço em disco e uma placa de rede em modo *Bridge* para permitir a comunicação entre as máquinas na mesma faixa de endereço IP da rede local.

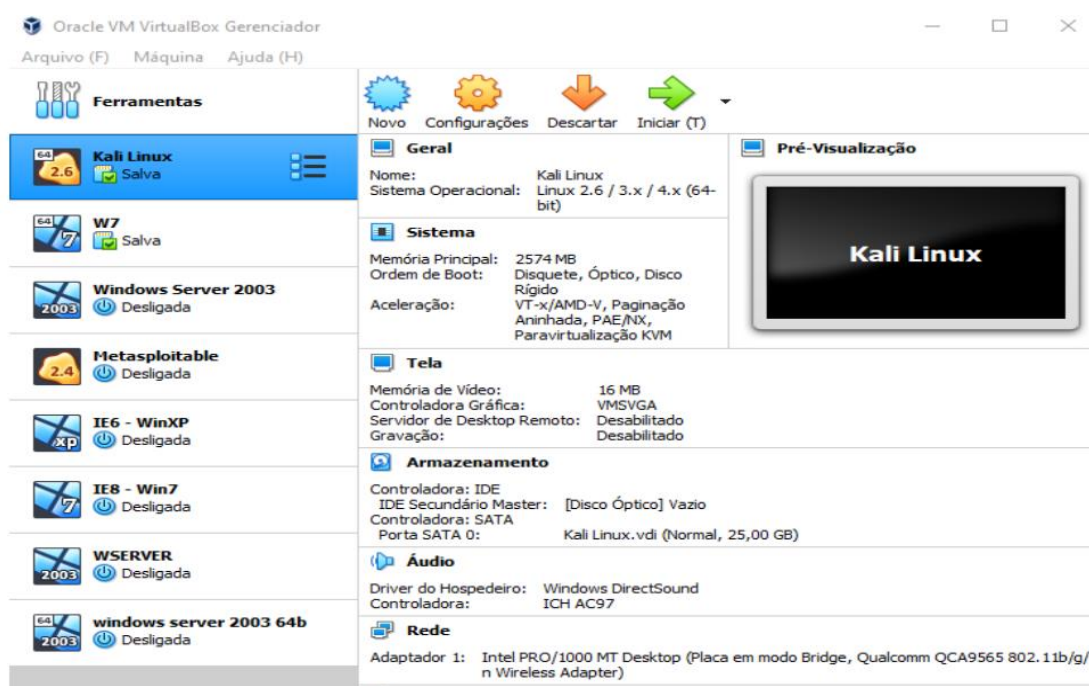


Figura 3 - *Virtualbox* com as máquinas virtuais. Fonte: ASSUNÇÃO, 2017.

3. REFERENCIAL TEÓRICO

3.1 - ETAPAS DO PENTEST

Como já abordado anteriormente, o *Penetration Test (Pentest)* é uma técnica utilizada para avaliar a segurança de um sistema de computador ou rede, simulando um ataque de um *hacker* para identificar vulnerabilidades e falhas de segurança que possam ser exploradas. Diante disso, o *Pentest* é realizado em várias etapas, desde o planejamento até a entrega do relatório final.



Figura 3 - Esquemático das etapas de um teste de vulnerabilidades. Fonte: Ali e Heriyanto, 2011.

3.1.1 - Planejamento

A primeira etapa do *Pentest* é o planejamento, que envolve a definição dos objetivos do teste, a identificação do escopo e a obtenção de autorização do cliente para realizar o teste. Segundo Calderón (2015), o planejamento é essencial para o sucesso do *Pentest*, pois permite que a equipe de teste entenda os objetivos do cliente e identifique as áreas críticas a serem testadas. Nessa etapa, é importante considerar os possíveis impactos e riscos para o sistema e o ambiente de produção, para evitar interrupções não planejadas.

3.1.2 - Coleta de informações

A etapa de coleta de informações é essencial para identificar as possíveis vulnerabilidades do sistema. Nessa etapa, são coletados dados sobre a infraestrutura, sistemas operacionais, aplicativos, serviços e usuários. De acordo com Singh e Arora (2021), a coleta de informações pode ser realizada por meio de técnicas passivas, como a análise de informações disponíveis publicamente, ou por meio de técnicas ativas, como a varredura de portas e serviços abertos.

3.1.3 - Análise de vulnerabilidades

A etapa de análise de vulnerabilidades envolve a identificação de vulnerabilidades e falhas de segurança no sistema ou rede testada. Essa etapa é realizada por meio de ferramentas de análise de vulnerabilidades e testes manuais. De acordo com Zouridaki e Antonatos (2016), é importante que os testes manuais sejam realizados por especialistas em segurança da informação, que possam identificar vulnerabilidades que não seriam detectadas por ferramentas automatizadas.

3.1.4 - Exploração de vulnerabilidades

Na etapa de exploração de vulnerabilidades, as vulnerabilidades identificadas na etapa anterior são exploradas para avaliar o impacto de um ataque e verificar se é possível invadir o sistema ou rede testada. De acordo com Bhardwaj e Singh (2017), essa etapa deve ser realizada com cautela e sempre com autorização do cliente, para evitar danos ao sistema ou ambiente de produção.

3.1.5 - Pós - exploração

Após a exploração de vulnerabilidades, é realizada a etapa de pós-exploração, que envolve a obtenção de informações adicionais sobre o sistema ou rede testada. Essas informações podem incluir dados sobre a configuração do sistema, usuários e permissões de acesso, entre outros. De acordo com Sood e Anand (2017), essa etapa é importante para entender o sistema testado e identificar outras possíveis vulnerabilidades que possam ter sido ignoradas nas etapas anteriores.

3.1.6 - Relatório e recomendações

Por fim, a etapa de relatório e recomendações envolve a elaboração de um relatório detalhado sobre o *Pentest*, incluindo as vulnerabilidades encontradas, os riscos associados a cada vulnerabilidade e as recomendações para corrigir as falhas de segurança identificadas. De acordo com Pinto e Fernandes (2018), a elaboração do relatório de *Pentest* é uma das etapas mais críticas do processo, pois é o resultado final do trabalho realizado pela equipe de segurança. O relatório deve ser entregue ao cliente de forma clara e objetiva, com uma linguagem acessível para que o cliente possa compreender facilmente as informações apresentadas.

Além disso, o relatório deve ser elaborado de forma que as informações possam ser facilmente organizadas e localizadas, permitindo que o cliente possa tomar as medidas necessárias para corrigir as vulnerabilidades identificadas. Segundo Singh e Arora (2021), o relatório deve ser entregue ao cliente o mais breve possível após a realização do teste, permitindo que as correções possam ser realizadas rapidamente.

Por fim, é importante ressaltar que o relatório de *Pentest* deve ser confidencial e não pode ser divulgado a terceiros sem a autorização expressa do cliente. O objetivo do relatório é ajudar o cliente a aprimorar a segurança de seu sistema ou rede, não expor suas vulnerabilidades a terceiros que possam utilizá-las para fins maliciosos.

3.2 - BACKTRACK

BackTrack é um sistema operacional baseado no Ubuntu que é amplamente utilizado em testes de penetração e auditorias de segurança. Ele é conhecido por suas avançadas ferramentas que permitem identificar, detectar e explorar vulnerabilidades em sistemas alvo, emulando assim vários tipos de ataques (Ali e Heriyanto, 2011).

O *BackTrack* possui uma variedade de ferramentas, incluindo coleta de informações, mapeamento de rede, identificação de vulnerabilidades, invasão, elevação de privilégios, manutenção de acesso, eliminação de rastros, analisadores de redes via rádio, análise de VoIP (*Voice Over Internet Protocol*) e telefonia, perícia forense digital e ferramentas para engenharia reversa. Diversas dessas ferramentas podem ser utilizadas em um projeto, como o OpenVAS, Metasploit Framework e o SET (*Social Engineering Toolkit*), que podem ser usados exaustivamente durante as fases de identificação de vulnerabilidades e realização de ataques.

3.2.1 - OpenVAS

OpenVAS é uma plataforma de gerenciamento de vulnerabilidades que opera em uma arquitetura cliente-servidor, permitindo a seleção de alvos na rede para testes de varreduras de vulnerabilidades. Possui vários componentes, como o OpenVAS Scanner, OpenVAS Client, OpenVAS Manager, Greenbone Security Assistant (GSA) e OpenVAS Administrator. O OpenVAS Scanner gerencia a execução dos testes de vulnerabilidade de rede, o OpenVAS Client controla a execução da varredura no sistema alvo, o OpenVAS Manager armazena os resultados das varreduras e executa funções como agendamento de tarefas e geração de relatórios. O GSA é uma interface web para configurar e gerenciar o processo de varredura, enquanto o OpenVAS Administrator é responsável pela administração e logs dos usuários.

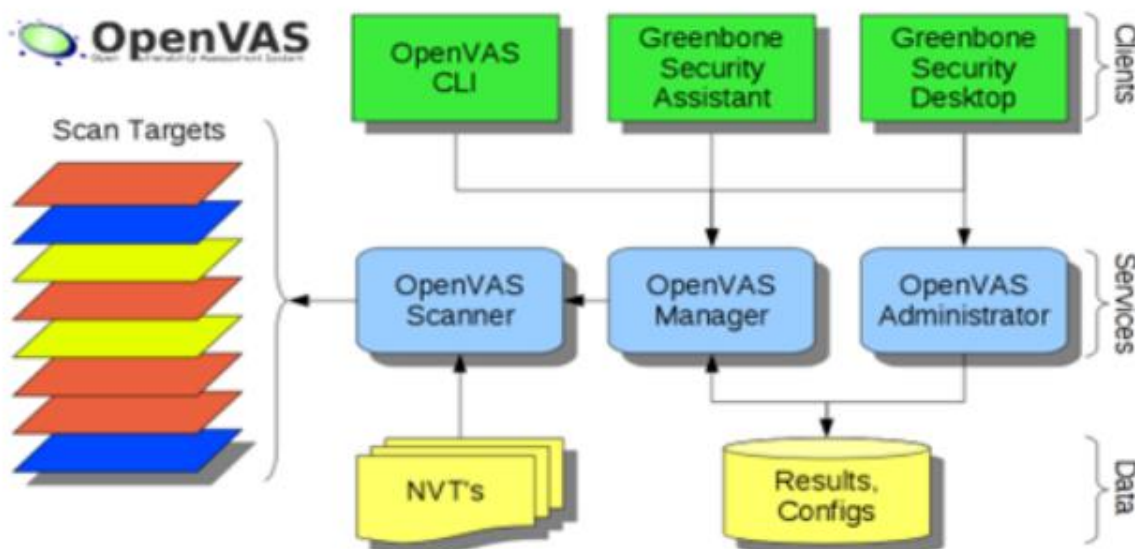
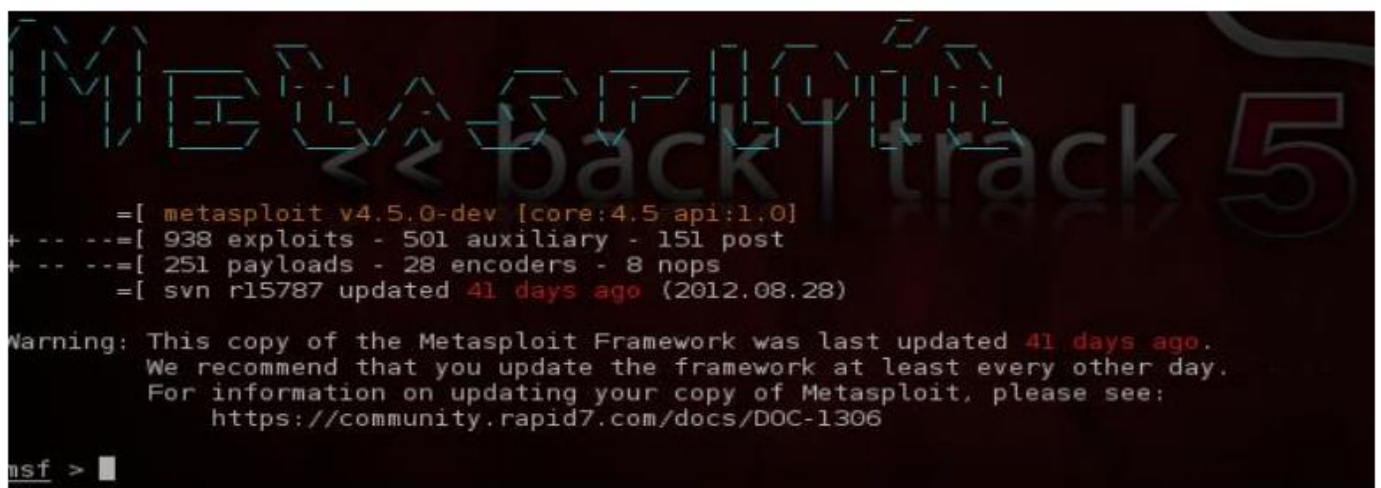


Figura 4 - Esquemático de componentes do OpenVAS . Fonte: www.openvas.org

3.2.2 - Metasploit Framework

O Metasploit Framework é uma plataforma *open source* que auxilia no desenvolvimento, teste e exploração de vulnerabilidades de software por meio de *exploits*. Esses *exploits* são softwares que exploram *bugs* ou vulnerabilidades presentes em um sistema, permitindo que o invasor realize ações, como burlar acessos restritos. O Metasploit Framework possui centenas de *exploits*, *payloads* e ferramentas avançadas para testar vulnerabilidades em diferentes plataformas e sistemas operacionais. Além disso, o Metasploit possui outras ferramentas, como os auxiliares e os encoders. Existem três interfaces de uso: MSFCLI, MSFWeb e MSFConsole, sendo esta última a mais eficiente e poderosa para um teste de vulnerabilidade. Durante um ataque utilizando o MSFConsole, a seleção do *exploit* deve ser feita tendo em mãos algumas informações do sistema alvo. O banco de dados do Metasploit também pode ser consultado para verificar a disponibilidade de um *exploit* para o serviço alvejado.



```
Metasploit v4.5.0-dev [core:4.5 api:1.0]
+ -- --[ 938 exploits - 501 auxiliary - 151 post
+ -- --[ 251 payloads - 28 encoders - 8 nops
+ -- --[ svn r15787 updated 41 days ago (2012.08.28)

Warning: This copy of the Metasploit Framework was last updated 41 days ago.
We recommend that you update the framework at least every other day.
For information on updating your copy of Metasploit, please see:
https://community.rapid7.com/docs/DOC-1306

msf > █
```

Figura 5 - Tela inicial do Metasploit Framework . Fonte: Autoral.

O processo de invasão começa com a escolha do *exploit* e do alvo, seguido pela seleção do *payload*. Cada *exploit* tem seus próprios *payloads*, que são selecionados com base na vulnerabilidade a ser explorada. Depois de configurar as opções do *payload*, o *exploit* é executado. Em alguns casos, há uma fase de pós-exploração, que pode ser realizada com o uso do Meterpreter. O Meterpreter é uma plataforma de exploração que oferece uma variedade de atividades possíveis de serem executadas no sistema explorado, incluindo a elevação de privilégios, *keylogging* e criação de um *backdoor* persistente. Ele também permite a migração de processos, o que ajuda a "camuflar" a invasão e dificultar a detecção.

3.2.3 - Ferramenta de engenharia social

O *Social Engineer Toolkit* (SET) é uma ferramenta do *BackTrack* utilizada para ataques de engenharia social, que exploram fraquezas humanas como curiosidade, falta de conhecimento ou descuido. Os ataques do SET usam vetores, que fornecem acesso ou informação do sistema. O *SphearPhishing Attack Vector*, um dos principais ataques oferecidos pelo SET, consiste em criar um arquivo de *exploit* em formato de arquivo, como PDF, e enviá-lo por e-mail para a vítima. Quando aberto, o arquivo compromete o sistema alvo.

```
The Spearphishing module allows you to specially craft email messages and send them to a large (or small) number of people with attached fileformat malicious payloads. If you want to spoof your email address, be sure "Sendmail" is installed (apt-get install sendmail) and change the config/set_config SENDMAIL=OFF flag to SENDMAIL=ON.

There are two options, one is getting your feet wet and letting SET do everything for you (option 1), the second is to create your own FileFormat payload and use it in your own attack. Either way, good luck and enjoy!

1) Perform a Mass Email Attack
2) Create a FileFormat Payload
3) Create a Social-Engineering Template

99) Return to Main Menu

set:phishing>
```

Figura 6 - Tela do exploit Sphearphising. Fonte: Autoral.

Os ataques web são os mais completos e incluem o *Java Applet*, que se adapta ao *browser* da vítima e entrega um *payload* disfarçado como serviço autêntico, o *Metasploit Browser Exploit*, que usa vulnerabilidades específicas do *browser*, e o *Credentials Harvesting*, que clona um site como o Gmail para obter as credenciais de login do usuário.

```
1) Java Applet Attack Method
2) Metasploit Browser Exploit Method
3) Credential Harvester Attack Method
4) Tabnabbing Attack Method
5) Man Left in the Middle Attack Method
6) Web Jacking Attack Method
7) Multi-Attack Web Method
8) Victim Web Profiler
9) Create or import a CodeSigning Certificate

99) Return to Main Menu

set:webattack>
```

Figura 7 - Tela do exploit Website Attack Vectors. Fonte: Autoral.

4. CONCLUSÃO

Ao longo das discussões realizadas neste presente artigo, foi possível abordar diversos tópicos relacionados a *Pentest*, desde conceitos básicos até ferramentas e técnicas utilizadas em ataques. Ficou evidente a importância do *Pentest* como uma técnica preventiva de segurança da informação, permitindo a identificação de vulnerabilidades em sistemas e redes antes que elas sejam exploradas por invasores mal-intencionados.

Além disso, destacou-se a necessidade de profissionais qualificados e especializados em *Pentest*, capazes de realizar testes de forma ética e responsável, seguindo as normas e regulamentações vigentes. Também foram discutidos aspectos legais e éticos envolvidos em testes de invasão, como a obtenção de autorização prévia dos proprietários dos sistemas testados.

Por fim, foi ressaltado o papel das ferramentas de *Pentest*, como o *BackTrack* e o *Social Engineer Toolkit* (SET), que permitem aos profissionais de segurança da informação realizar testes de invasão de forma mais eficiente e abrangente. No entanto, é importante lembrar que a eficácia dessas ferramentas depende do conhecimento e habilidade do profissional que as utiliza.

Em suma, o *Pentest* é uma técnica de segurança da informação fundamental para garantir a proteção dos sistemas e redes de empresas e organizações. É importante que as empresas entendam sua importância e invistam em profissionais qualificados e em ferramentas especializadas, a fim de garantir a segurança de suas informações e dados sensíveis.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- ALI, R. M.; HERIYANTO, E. A. **Backtrack as a Penetration Testing Tool: An Overview**. 2011 International Conference on Electrical Engineering and Informatics, Bandung, Indonesia, 2011. Proceedings... Piscataway: IEEE, 2011. p. 1-4. Acesso em: 13 de abril de 2023.
- ALLSOPP, Will. **Advanced penetration testing: hacking the world 's most secure networks**. Indianapolis: Wiley, 2017. Acesso em: 30 de março de 2023.
- ASSUNÇÃO, Marcos Flávio Araújo. Segredos do Hacker Ético. 5. ed. Florianópolis: Editora Visual Books, 2014. Acesso em: 11 de abril de 2023.
- BERTOGLIO, Daniel Dalalana; ZORZO, Avelino Francisco. **Um Mapeamento Sistemático sobre Testes de Penetração**. 2015. 42 f. Monografia (Doutorado) - Curso de Pós-Graduação em Ciência da Computação, Pontifícia Universidade Católica do Rio Grande do Sul Faculdade de Informática, Porto Alegre, 2015. Acesso em: 11 de abril de 2023.

- BHARDWAJ, R., & SINGH, J. P. An approach to penetration testing of cloud computing infrastructure. In 2017 IEEE 7th International Advance Computing Conference (IACC) (pp. 145-150). IEEE. 2017. Acesso em: 12 de abril de 2023.
- CALDERÓN, J. M. **The penetration testing execution standard 2.0**. The SANS Institute, 2015. Acesso em: 12 de abril de 2023.
- KENNEDY, David et al. **Metasploit: The Penetration Tester 's Guide**. No Starch Press, 2011. Acesso em: 30 de março de 2023.
- PINTO, L. M.; FERNANDES, R. M. **Um estudo sobre as etapas do Pentest e a importância do relatório de testes**. Revista Brasileira de Computação Aplicada, v. 10, n. 1, p. 1-12, 2018. Acesso em: 12 de abril de 2023.
- SINGH, A., & ARORA, S. **A comprehensive study of penetration testing tools, techniques, and methodologies**. International Journal of Engineering Research & Technology, 10(3), 464-468. 2021. Acesso em: 12 de abril de 2023.
- SOOD, A. K., & ANAND, A. **Comprehensive approach to penetration testing**. 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICIC) (pp. 1-6). IEEE. Acesso em: 12 de abril de 2023.
- STUTTARD, Dafydd; PINTO, Marcus. **The Web Application Hacker 's Handbook: Finding and Exploiting Security Flaws**. 2nd ed. Indianapolis: Wiley Publishing, 2011. Acesso em: 29 de março de 2023.
- VERIZON. **Data Breach Investigations Report**. 2020. Acesso em: 29 de março de 2023. Disponível em: <<https://enterprise.verizon.com/resources/reports/2020-data-breach-investigations-report.pdf>>.
- WEIDMAN, Georgia. **Penetration Testing: A Hands-On Introduction to Hacking**. No Starch Press, 2014. Acesso em: 30 de março de 2023.
- ZOURIDAKI, C., & ANTONATOS, S. **A survey of penetration testing methodologies in cloud environments**. Future Internet, 8(4), 51. 2016. Acesso em: 12 de abril de 2023.