1.Introdução

Para entender os aspectos técnicos do hacking ético, é preciso entender mais sobre quais são as responsabilidades do trabalho de um pentester e quais processos são seguidos na realização de testes de invasão.

A importância e relevância da segurança cibernética estão cada vez maiores e podem estar presentes em todas as esferas da vida. A segurança cibernética é relevante para todas as pessoas no mundo moderno, incluindo uma política de senha forte para proteger seus e-mails ou para empresas e outras organizações que precisam proteger dispositivos e dados contra danos.

Um teste de penetração é uma tentativa ética de testar e ensinar as defesas de segurança para proteger esses ativos e informações. O teste envolve o uso das mesmas ferramentas, técnicas e metodologias que alguém com intenções maliciosas usaria e é semelhante a uma auditoria

Rótulos como “hacking” e “hacker” costumam ter conotações negativas, especialmente na cultura pop. A ideia de obter acesso legal a um sistema informático é um conceito difícil de compreender. Um teste de penetração é uma auditoria autorizada da segurança e das defesas de um sistema de computador, conforme acordado pelos proprietários dos sistemas. Antes de iniciar um teste de penetração, ocorre uma discussão formal entre o testador de penetração e o proprietário do sistema. Várias ferramentas, técnicas e sistemas a serem testados são acordados. Essa discussão constitui o **escopo do acordo de teste de penetração** e irá determinar o curso que o teste de penetração seguirá.

As empresas que fornecem serviços de testes de penetração são responsabilizadas pelas estruturas legais e pelo credenciamento do setor. A ética é o debate moral entre o certo e o errado, onde uma ação pode ser legal, ela pode ir contra o sistemas de crenças de certo e errado de um indivíduo. Os testadores de penetração muitas vezes irão se deparar com decisões potencialmente questionáveis do ponto de vista moral durante um teste de penetração.

Os hackers são classificados em três categorias, onde sua ética e motivação por trás de suas ações determinam em que categoria eles são colocados.

Hat	Descrição	Exemplo
White hat	Hackers considerados “pessoas boas”. Permanecem dentro da lei e usam suas habilidades para beneficiar outros	Um testador de penetração realizando um trabalho autorizado em uma empresa
Grey hat	Essas pessoas costumam usar suas habilidades para beneficiar outras pessoas; no entanto, eles não respeitam/seguem a lei ou os padrões	Alguém derrubando um site fraudulento

	éticos em todos os momentos	
Black hat	Estas pessoas são criminosas e muitas vezes procuram prejudicar organizações ou obter alguma forma de benefício financeiro à custa de outros	Os autores de ransomware infectam dispositivos com código malicioso e retêm dados para resgate

1.1 Rules of Engagement

O **ROE** (*Rules of Engagement*) é um documento criado nos estágios iniciais de um trabalho de teste de penetração. Este documento consiste em três seções principais que são responsáveis, em última instância, por decidir como o trabalho será realizado. O instituto SANS tem um ótimo exemplo no link que segue: https://sansorg.egnyte.com/dl/bF4I3yCcnt/?

Seção	Descrição
Permissão	Esta seção do documento dá permissão explícita para que o trabalho seja realizado. Esta permissão é essencial para proteger legalmente indivíduos e organizações pelas atividades que realizam
Escopo de teste	Esta seção do documento irá anotar metas específicas às quais o compromisso deve ser aplicado. Por exemplo, o teste de penetração pode aplicar-se apenas a determinados servidores ou aplicações, mas não a toda rede
Regras	A seção de regras definirá exatamente as técnicas permitidas durante o engajamento. Por exemplo, as regras podem estabelecer especificamente que técnicas como ataques de phishing são proibidas, mas ataques MitM são aceitáveis

Os testes de penetração podem ter uma ampla variedade de objetivos e metas dentro do escopo. Por causa disso, nenhum teste de penetração é igual e não existe um caso único de como abordá-lo.

As etapas que um testador de penetração executa durante um envolvimento são conhecidas como metodologia. Uma metodologia prática é inteligente, onde as medidas tomadas são relevantes para a situação em questão. Abaixo, estão uma descrição geral destas etapas.

Estágio	Descrição
Coleta de informações	Esta fase envolve a recolha do máximo possível de informações publicamente acessíveis sobre um alvo/organização, como OSINT
Enumeração/Verificação	Esta etapa envolve a descoberta de aplicativos e serviços em execução nos sistema, Por exemplo, encontrar um servidor web que possa ser potencialmente vulnerável
Exploração	Este estágio envolve aproveitar as vulnerabilidades descobertas em um sistema ou aplicativo. Aqui, pode envolver

	o uso de exploração pública ou a exploração lógica do aplicativo
Escalação de privilégios	Depois de explorar com sucesso um sistema ou aplicativo, este estágio é a tentativa de expandir seu acesso a um sistema. É possível escalar horizontalmente ou verticalmente, onde horizontal é acessar outra conta do mesmo grupo de permissão e vertical é o de outro grupo de permissão (administrador)
Pós-exploração	Quais outros hosts podem ser direcionados? Que informações adicionais podemos coletar do host agora que somos um usuário privilegiado? Cobrindo seus rastros Relatórios

1.2 Open Source Security Testing Methodology Manual

O **OSSTMM** (*Open Source Security Testing Methodology Manual*) fornece uma estrutura detalhada de estratégias de teste para sistemas, software, aplicativos, comunicações e o aspecto humano da segurança cibernética. Segue o link: https://www.isecom.org/OSSTMM.3.pdf

A metodologia centra-se principalmente na forma como estes sistemas e aplicações comunicam, pelo que inclui uma metodologia para: telecomunicações (telefones, VoIP), redes com fio, comunicações sem fio.

Vantagens	Desvantagens
Abrange várias estratégias de teste em profundidade	A estrutura é difícil de entender, muito detalhada e tende a usar definições únicas
Inclui estratégias de teste para alvos específicos	
A estrutura é flexível dependendo das necessidades da organização	
A estrutura pretende estabelecer um padrão para sistemas e aplicações, o que significa que uma metodologia universal pode ser usada em um cenário de testes de penetração	

1.3 Open Web Application Security Project

A **OWASP** (*Open Web Application Security Project*) é uma estrutura dirigida pela comunidade e frequentemente atualizada, usada exclusivamente para testar segurança de aplicações e serviços da web. Segue o link: https://owasp.org/

A fundação escreve regularmente relatórios informando as dez principais vulnerabilidades de segurança que um aplicativo da web pode ter, a abordagem de teste e a correção.

Vantagens	Desvantagens
Fácil de entender	Pode não estar claro que tipo de vulnerabilidade um aplicativo da web possui
Mantido ativamente e atualizado com frequência	A OWASP não faz sugestões para nenhum ciclo de vida de desenvolvimento de software específico
Abrange todas as fases de um trabalho: desde testes até relatórios e remediação	A estrutura não possui nenhum credenciamento como CHECK
Especializada em aplicações e serviços web	

1.4 National Institute of Standard Technology

A estrutura cibernética do **NIST** é uma estrutura popular usada para melhorar os padrões de segurança cibernética de uma organização e gerenciar o risco de ameaças cibernéticas. Ela fornece diretrizes sobre controles de segurança e benchmarks para o sucesso de organizações, desde infraestrutura crítica até comercial. Há uma seção limitada sobre uma diretriz padrão para a metodologia que um testador de penetração deve seguir.

Vantagens	Desvantagens
Estima-se que o NIST framework seja usado por 50% das organizações americanas até 2020	O NIST tem muitas iterações de estruturas, por isso pode ser difícil decidir qual delas se aplica à sua organização
A estrutura é extremamente detalhada no estabelecimento de padrões para ajudar as organizações a mitigar a ameaça representada pelas ameaças cibernéticas	A estrutura do NIST possui políticas de auditorias fracas, tornando difícil determinar como ocorreu uma violação
A estrutura é atualizada com muita frequência	A estrutura não considera a computação em nuvem, que está rapidamente se tornando cada vez mais popular para as organizações
O NIST fornece credenciamento para organizações que usam esta estrutura	
A estrutura do NIST foi projetada para ser implementada juntamente com outras estruturas	

1.5 Cyber Assessment Framework

O **NCSC CAF** (*Cyber Assessment Framework*) é uma estrutura extensa de quatorze princípios usados para avaliar o risco de várias ameaças cibernéticas e as defesas de uma organização contra elas. A estrutura se aplica a organizações

consideradas para realizar “serviços e atividades de vital importância”, como infraestrutura crítica, serviços bancários e similares. A estrutura concentra-se e avalia principalmente os seguintes tópicos:

- Segurança de dados
- Sistemas de segurança
- Controle de identidade e acesso
- Resiliência
- Monitoramento
- Planejamento de resposta e recuperação

Vantagens	Desvantagens
Esta estrutura é apoiada por uma agência governamental de segurança cibernética	A estrutura ainda é nova na indústria, o que significa que as organizações não tiveram muito tempo para fazer as mudanças necessárias para se adequarem a ela
Esta estrutura fornece acreditação	A estrutura é baseada em princípios e ideias e não é tão direta quanto ter regras como algumas outras estruturas
Este quadro abrange quatorze princípios que vão desde a segurança até à resposta	

2.Boxes

Existem três escopos principais ao testar um aplicativo ou serviço. Sua compreensão do seu alvo determinará o nível de teste que você realizará em seu envolvimento de teste de penetração.

2.1 Black box

Este processo de teste é um processo de alto nível em que o testador não recebe nenhuma informação sobre o funcionamento interno do aplicativo ou serviço. O testador atua como um usuário regular testando a funcionalidade e a interação do aplicativo ou software. Este teste pode envolver a interação com a interface, ou seja, botões, e testes para ver se o resultado pretendido é retornado. Nenhum conhecimento de programação ou compreensão do programa é necessário para este tipo de teste.

Os testes **black box** aumentam significativamente a quantidade de tempo gasto durante a fase de coleta e enumeração de informações para compreender a superfície de ataque do alvo

2.2 Grey box

Este processo de teste é o mais popular para coisas como testes de penetração. É uma combinação de processos de teste de **black box** e **white box**. O testador terá algum conhecimento limitado dos componentes internos do aplicativo ou software. Ainda assim, ele interagir com o aplicativo como se fosse um cenário de **black box**, e em seguida, usará seu conhecimento do aplicativo para tentar resolver os problemas à medida que os encontrar.

2.3 White box

Este processo de teste é um processo de baixo nível, geralmente feito por um desenvolvedor de software que conhece programação e lógica de aplicação. O testador testará os componentes internos do aplicativo ou software, e, por exemplo, garantirá que funções específicas funcionem corretamente e dentro de um período de tempo razoável.

O testador terá pleno conhecimento do aplicativo e de seu comportamento esperado e consome muito mais tempo do que o teste de **black box**. O conhecimento completo em um cenário de teste de **white box** fornece uma abordagem de teste que garante que toda a superfície de ataque possa ser validada.

3.The Hacker Methodology

A primeira etapa da metodologia do hacker é a **reconnaissance**, ou reconhecimento. O reconhecimento consiste na coleta de informações sobre seu alvo e de modo geral, não envolve interação com o(s) alvo(s) ou sistemas. É um conceito bastante simples, pense em quais ferramentas podemos usar na Internet para coletar informações sobre as pessoas.

O **Google** é uma ferramenta incrivelmente útil e existem maneiras de usar de forma eficaz. Também é possível utilizar sites como a **Wikipedia** para entender sobre diversos assuntos, **Youtube** e até mesmo perfis no **LinkedIn**. Mas é bom ter em conhecimento as seguintes ferramentas: **Google** (google dorking), **Wikipedia**, **PeopleFinder.com**, **who.is**, **sublist3r**, **hunter.io**, **buildwith.com**, **wappalyzer**.

A segunda fase da metodologia é **Scanning** e **Enumeration**. É nesta parte onde o hacker irá começar a interagir com o alvo para tentar encontrar vulnerabilidades relacionadas ao alvo. Ferramentas mais especializadas começam a se tornar necessário tais como: **nmap**, **dirb**, **metasploit**, **exploit-db**, **Burp Suite** e outras que são úteis para ajudar a tentar encontrar vulnerabilidades em um alvo. Na fase de **Scanning**, o invasor interage com o alvo para determinar sua **superfície geral de ataque**. Essa superfície de ataque determina a que alvo pode estar vulnerável na fase de exploração. Essas vulnerabilidades podem ser uma variedade de coisas como: a página da web que não está devidamente bloqueada, um site vazando informações, injeções de SQL, scripts entre sites (XSS) e outras vulnerabilidades.

A fase de **exploração** de um pentest não é tão glamorosa quanto parece. Ela só pode ser tão boa quanto as fases de reconhecimento e enumeração. Se você não enumerar todas as vulnerabilidades, poderá perder uma oportunidade ou se não olhar com atenção o alvo, a exploração que você escolheu pode falhar completamente.

Uma ferramenta comum usada para exploração é chamada de **Metasploit**, que possui muitos scripts integrados para tentar manter a vida simples. Também é possível utilizar ferramentas como **Burp Suite** e **SQLMap** para explorar aplicativos da web. Existem ferramentas como **msfvenom** e **BeEF** e outras.

Depois de se obter acesso a uma máquina vítima através da **fase de exploração**, o próximo passo é **escalar os privilégios** para uma conta de usuário superior. Os seguintes relatos são o que tentamos alcançar como pentest: no mundo **Windows**, a conta de destino geralmente é **administrador** ou **system**; no mundo **Linux**, a conta de destino geralmente é **root**.

Em qualquer sistema operacional um dispositivo está sendo executado é muito importante para determinar como se aumentar o privilégio posteriormente. Assim que obtiver acesso como usuário de nível inferior, tentar executar outro exploit ou encontrar uma maneira de se tornar **root** ou **administrador**.

O escalonamento de privilégios pode assumir muitas formas, alguns exemplos são:

- Quebrando hashes de senha encontrados no alvo
- Encontrar um serviço vulnerável ou versão de um serviço que permitirá escalar privilégios através do serviço
- Espalhamento de senha de credenciais descobertas anteriormente
- Usando credenciais padrão
- Encontrar chaves secretas ou chaves SSH armazenadas em um dispositivo que permitirá a migração para outra máquina
- Executar scripts ou comandos para enumerar configurações do sistema como ‘ifconfig’ para encontrar configurações de rede ou o comando ‘find / -perm’

A maioria dos pentesters profissionais/éticos nunca tem a necessidade de **cobrir seus rastros**, no entanto, esta ainda é uma fase da metodologia. Você deve sempre ter permissão explícita do proprietário do sistema sobre quando o teste está acontecendo, como está ocorrendo e o escopo dos alvos em qualquer teste de penetração.

Como as regras de engajamento para um teste de penetração devem ser acordadas antes do teste ocorrer, o testador de penetração deve parar imediatamente

quando tiver alcançado o escalonamento de privilégios e relatar a descoberta ao cliente.

Como tal, um profissional nunca irá cobrir os seus rastros porque a avaliação foi planejada e acordada previamente. Mesmo que você cubra seus rastros, isso não o isenta da responsabilidade por sua exploração. Frequentemente, você precisará ajudar o administrador de TI ou proprietário do sistema a limpar o código de exploração utilizado e também recomendar como prevenir o ataque no futuro.

Embora os hackers éticos raramente precisem encobrir seus rastros, você ainda deve monitorar e anotar cuidadosamente todas as tarefas executadas como parte do teste de penetração para ajudar a corrigir as vulnerabilidades e recomendar alterações ao proprietário do sistema.

A fase final da metodologia pentest é a **fase de relatórios**. Esta é uma das fases mais importantes onde você irá delinear tudo o que encontrou. Geralmente essa fase inclui o seguinte: as descobertas ou vulnerabilidades, a criticalidade da descoberta, uma descrição ou breve visão geral de como a descoberta foi descoberta e recomendações de correção para resolver a descoberta.

A quantidade de documentação de relatório varia amplamente de acordo com o tipo de envolvimento em que o pentester está envolvido. Um relatório de descobertas geralmente vem em três formatos: Resultados da verificação de vulnerabilidades; Resumo das descobertas; Relatório formal completo.