Análise de vulnerabilidades e dos tipos de

Descrição

Descrição de mapeamento de redes, vulnerabilidades e tipos de ataques a pessoas e instituições. Demonstração do procedimento de ataque e exemplificação de formas de ataques diferentes.

Propósito

O conhecimento de diferentes formas de ataque é essencial para um técnico de TI manter a segurança em redes, serviços, aplicações e até para pessoas no meio digital ou aqueles que escolherem trabalhar com segurança da informação ou não.

Objetivos

Introdução

O presente conteúdo tem como objetivo demonstrar e analisar os principais **métodos de ataques** a pessoas e instituições. Inicialmente, será discutido como um invasor ou atacante consegue informações sobre uma rede. Esses processos fazem parte da análise inicial de um *hacker* para conhecer a rede alvo.

Em seguida, serão discutidos os principais tipos de ataques utilizados por invasores para tentar comprometer uma rede, causando danos aos serviços instalados nessa rede e às pessoas que a utilizam. Será dada atenção especial a códigos maliciosos, ou *malwares*, e ataques a redes sem fio, no fim do texto.

Todos os procedimentos abordados aqui podem ser utilizados pelos atacantes, mas também pela equipe de TI responsável pela rede e pelos serviços, interna e externamente a uma instituição. Esses conhecimentos são base para manter a segurança dessas redes e também das pessoas que a utilizam. Isso porque, para se defender, é importante saber como será atacado.

Mapeamento de Redes

Ao final deste módulo, você será capaz de relacionar os principais meios de varredura de redes e vulnerabilidades.

O que é mapeamento de redes?

Definição

O **mapeamento de redes** pode ser compreendido como o processo de descoberta de hosts ativos, sistemas operacionais desses hosts, serviços abertos e tudo que for possível descobrir sobre determinada rede. Para isso, são utilizadas ferramentas de manipulação de pacotes *TCP*, *UDP*, *ICMP*.

Esse conjunto de técnicas é utilizado por atacantes para tomar conhecimento da estrutura da rede e como funciona.

Aplicações

Atualmente, o mapeamento de redes é parte importante no processo de *Penetration Test*, ou *pentest*, em que uma equipe é contratada para simular o ataque de um agente malicioso em sua rede.

O mapeamento de rede é uma das primeiras fases desse processo, para identificar os servidores e aplicações que estão disponíveis na rede.

Fases

O mapeamento de redes pode ser dividido em fases: *Host Scan*, *Port Scan*, *Service Scan/OS fingerprint* e **varredura de vulnerabilidade**.

A primeira fase, *Host Scan*, tem como objetivo descobrir quais são os servidores, máquinas ou dispositivos ligados à rede. Por organização, é importante executar o *host scan* antes de qualquer outra varredura.

Host Scan

A primeira fase tem como objetivo descobrir quais são os servidores, máquinas ou dispositivos ligados à rede.

Port Scan

Port Scan tem como objetivo descobrir quais são as portas TCP e UDP abertas nesses servidores.

Service Scan/OS fingerprint

Service Scan/OS fingerprint objetiva descobrir quais são serviços no servidor e qual o seu sistema operacional.

Varredura

A varredura de vulnerabilidades é a fase de pesquisa que indica quais vulnerabilidades os servidores possuem.

Por organização, é importante executar o *host scan* antes de qualquer outra varredura.

Hot Scan – ferramente *ping*

O utilitário *ping* é encontrado nativamente na maioria dos sistemas operacionais. Essa ferramenta utiliza o protocolo *ICMP (Internet Control Message Protocol* ou, em português, Protocolo de Mensagens de Controle da Internet).

O *ping* envia um pacote de *ICMP* Request para uma máquina. Se a máquina alvo estiver ativa e respondendo pacotes *ICMP*, ela responderá com um pacote *ICMP Reply*.

A execução do ping é muito simples e é demonstrada na imagem a seguir.

Captura de tela da ferramenta de linha comando ping.

Cada linha que se inicia com “64 bytes” representa um pacote *ICMP Reply* recebido pela máquina alvo que, no caso, é 8.8.8.8.

Também é possível realizar o *Host Scan* com a ferramenta fping, que envia pacotes *ICMP* para múltiplas máquinas simultaneamente. A imagem a seguir mostra um exemplo do comando executado pelo *fping*.

Captura de tela da ferramenta de linha comando *fping*.

Ferramenta nmap

O *nmap*, ou *Network Mapper*, é uma ferramenta extremamente poderosa quando se fala em mapeamento de redes. É capaz de verificar a conectividade entre *hosts*, verificar portas abertas em hosts remotos e, ainda, identificar os serviços e vulnerabilidades nas portas de hosts.

Isso é possível devido à utilização de várias técnicas de mapeamento de rede, além da utilização de *scripts* para mapeamento de serviços, vulnerabilidades e até mesmo para tentar identificar o sistema operacional de um *host* remoto.

Hot Scan – nmap

Até agora, foi mostrado como fazer o *Host Scan* com as ferramentas *ping* e *fping*. O *nmap* desempenha essa tarefa de uma forma mais completa, testando não apenas pacotes *ICMP* de diferentes tipos, mas também verificando as portas 443 e 80 do protocolo *TCP (Transport Layer Protocol*, ou, em português, Protocolo da Camada de Transporte).

Captura de tela de Host Scan com a ferramente nmap na rede 192.168.56.0/24.

O *nmap*, por padrão, primeiro descobre se o *host* está ativo e, caso esteja, executa uma varredura nas mil portas mais utilizadas. A opção '*-sn*' mostrada na imagem é utilizada para dizer ao nmap que ele deve executar apenas o teste se os *hosts* estão ativos. No resultado, é possível ver que o único host identificado como ativo pelo *nmap* é o 192.168.56.105.

A tabela a seguir mostra outras possibilidades de como escolher os endereços de ip a serem testados pela ferramenta.

Opção	Explicação
$ nmap -sn 192.168.56.105	Teste no endereço de ip 192.168.56.105
$ nmap -sn 192.168.56.0/24	Teste nos endereços de ip de 192.168.56.1 até 192.168.56.255.

Opção	Explicação
$ nmap -sn 192.168.56.1-100	Teste nos endereços de ip de 192.168.56.1 até 192.168.56.100.
$ nmap -sn -iL arquivo.txt	Teste nos endereços de ip escritos no arquivo.txt. O arquivo deve possuir um endereço de ip por linha.

Tabela – Opções do nmap para Host Scan
Fonte: Site da ferramenta nmap.org

Mapeamento de Portas com nmap

Como funciona um mapeamento de portas?

O *nmap* é capaz de mapear as portas *TCP* e *UDP* abertas em um ou mais servidores, processo conhecido como *Port Scan*.

A imagem, a seguir, mostra o resultado que a ferramenta nos mostra ao final da execução do *Port Scan* padrão, ou seja, da varredura das mil portas mais utilizadas, ou seja, portas comuns como 80 e 22, que representam os serviços de HTTP e SSH respectivamente.

Captura de tela da saída do Port Scan realizado com a ferramenta nmap.

Na primeira coluna, são mostrados os números das portas. Na segunda coluna, a palavra '*open*' indica que as portas estão abertas e, na última coluna, são listados os serviços associados a essas portas por padrão. A opção '*-Pn*' diz ao *nmap* para não realizar o *Host Scan*, pois já sabemos quais máquinas estão ativas na rede.

A tabela a seguir mostra outras opções relacionadas ao *Port Scan*.

Opção	Explicação
$ nmap -Pn 192.168.56.105 -p 22,80,443	Faz a varredura apenas nas portas 22,80 e 443.
$ nmap -Pn 192.168.56.105 -p 20-80	Faz a varredura em todas as portas de 20 até 80, inclusive.
$ nmap -Pn 192.168.56.105 -p-	Faz a varredura em todas as portas TCP, ou seja, da porta 1 até a porta 65535.
$ nmap -Pn 192.168.56.105 -p T:80,U:53	Faz a varredura na porta 80 TCP e na porta 53 UDP.
$ nmap -Pn 192.168.56.105 –top-ports=100	Faz a varredura nas 100 portas mais utilizadas no mundo.

Tabela – Opções do nmap para Port Scan

Port Scan padrão do nmap

O *Port Scan* padrão executado pelo *nmap* depende do usuário que está executando. Se for um usuário comum do sistema, será executado o *TCP Connect Scan*, e se for um usuário privilegiado, como *root* do Linux ou o Administrador do Windows, será executado o *TCP SYN Scan*.

3-Way Handshake

O *3-way handshake* é a etapa de estabelecimento de conexão do protocolo *TCP*. É chamada assim, pois há a troca de 3 pacotes no processo. Esse processo é importante, pois o *TCP* é um protocolo baseado em conexão, ou seja, ele realiza o controle de envio e recebimento de pacotes para garantir confiabilidade na conexão.

Esquema de 3-Way handshake do protocolo TCP.

Quando um cliente tenta iniciar uma conexão *TCP* com um servidor, um pacote *SYN (SYNchronize)* é enviado para alguma das 65535 portas do servidor. Esse pacote é chamado assim porque a *flag SYN* é marcada no pacote. As flags em um pacote TCP têm o objetivo de controlar a conexão. A *flag SYN* é utilizada quando uma conexão é iniciada.

Se a porta do servidor estiver aberta, ele responde com um pacote *SYN/ACK*. A *flag SYN* é marcada para iniciar a conexão e a *flag ACK* é marcada para informar ao cliente que o servidor acusa o recebimento do pacote anterior.

Em seguida, o cliente envia um pacote *ACK* para o servidor. Dessa forma, a conexão está estabelecida e a troca de dados será iniciada.

Port Scan com TCP Connect Scan

O *TCP Connect Scan* testa se uma porta está aberta, completando o *3-Way HandShake*. Se a porta estiver aberta, após o *3-Way Handshake*, o nmap envia um pacote *RST* para encerrar a conexão. Se a porta estiver fechada, o servidor responderá com um pacote *RST*. Dessa forma, o *nmap* é capaz de definir se a porta está aberta ou fechada.

Devido à quantidade de pacotes recebidos e enviados no processo, o *TCP Connect Scan* é considerado lento em relação a outros métodos. Além disso, é considerado menos furtivo, uma vez que completa as conexões TCP sem realizar nenhuma troca de dados.

Para forçar o *nmap* a utilizar esse método, deve-se usar a opção '*-sT*' na linha de comando.

Captura de tela da saída do TCP Connect Scan na porta 22 no endereço de ip 192.168.56.105.

Port Scan com TCP SYN Scan

O *TCP SYN Scan*, diferente do método anterior, não completa o *3-Way HandShake*. Nesse método, são enviados pacotes *SYN* para o servidor e o *nmap* aguarda a resposta do servidor. Se o nmap receber um pacote *SYN/ACK*, a porta está aberta, e se receber um pacote *RST*, a porta está fechada.

Processo de Port Scan com TCP SYN Scan quando a porta do servidor está aberta.

Por enviar apenas metade dos pacotes que o *TCP Connect Scan* enviaria, é bem mais rápido e furtivo.

Pode-se usar a opção '*-sS*' na linha de comando para obrigar o *nmap* a fazer a varredura com esse método.

Captura de tela da saída do TCP SYN Scan na porta 22 no endereço de ip 192.168.56.105.

Port Scan com UDP Scan

O *nmap* também é capaz de realizar varredura nas portas do protocolo *UDP (User Datagram Protocol*, ou Protocolo de Datagrama do Usuário). O *UDP Scan* é um método mais simples devido à pouca complexidade do protocolo. Basicamente, o *nmap* envia um pacote *UDP* sem dados para as portas a serem verificadas. Se o servidor não responder, a porta está aberta. Se o servidor responder com um pacote *ICMP* de erro, a porta está fechada.

Para executar essa varredura, pode-se utilizar a opção '*-sU*' na linha de comando, como mostra a figura a seguir.

Captura de tela da saída do UDP Scan na porta 53 no endereço de ip 192.168.56.105.

Mapeamento de vulnerabilidades

Service Scan

Depois de descobrir quais portas estão abertas nos *hosts*, é importante determinar quais serviços e quais as versões desses serviços. Com a opção '*-sV*', o *nmap* executa essa varredura.

Nesse método, são enviados pacotes para estabelecer a conexão com o *3-way handShake* e, em seguida, envia pacotes solicitando informações sobre os serviços.

A imagem a seguir mostra um exemplo de *Service Scan* com *nmap*.

Captura de tela da saída de Service Scan nas portas 21,22,23,80,139,445 no endereço de ip 192.168.56.105.

OS Fingerprint

O *nmap* também é capaz de tentar identificar qual é o sistema operacional de determinado servidor. Isso é possível devido a diferentes implementações em relação a redes e serviços em cada sistema operacional.

Um exemplo simples é o *TTL (Time to Live*, ou Tempo de Vida). Cada sistema possui uma definição do valor padrão de *TTL* para diferentes protocolos. A tabela a seguir mostra alguns exemplos importantes de *TTL* padrão de *ICMP*.

Sistema Operacional	TTL padrão
FreeBSD 5	64
Windows 10	128
Linux Kernel 2.4	255

Tabela – TTL de sistemas operacionais no protocolo ICMP.
Fonte: site do subinsb.com

Para que o *nmap* tente descobrir qual o sistema operacional de um alvo, pode-se utilizar a opção '*-O*'.

A descoberta do sistema operacional de um servidor é importante, pois o mesmo serviço disponibilizado por sistemas operacionais diferentes tem diferentes vulnerabilidades, já que são implementados de formas diferentes. Na hora de explorar as vulnerabilidades nos serviços, o atacante leva em consideração o sistema operacional do alvo.

Scripts do nmap

O *nmap* possui vários scripts com diferentes funções e categorias. Duas categorias importantes para a descoberta de vulnerabilidades são *default* e *vuln*.

Categoria default

Os *scripts* da categoria default são simples, de rápida execução e dão informações básicas sobre os serviços. Para executá-los, pode-se executar o *nmap* com a opção '*–script "default"*' ou com a opção '*-sC*'.

Categoria vuln

Os *scripts* da categoria *vuln* buscam, dentro dos serviços selecionados, vulnerabilidades conhecidas e fáceis de identificar, sem explorá-las. Para executar esses scripts, pode-se utilizar a opção '—*script "vuln"*'.

Captura de tela da saída de execução de scripts vuln da ferramenta nmap.

Ataques através de mapeamento de portas

Assista agora a um vídeo em que são apontados os principais ataques de mapeamento de portas e serviços em hosts ativos nas redes.

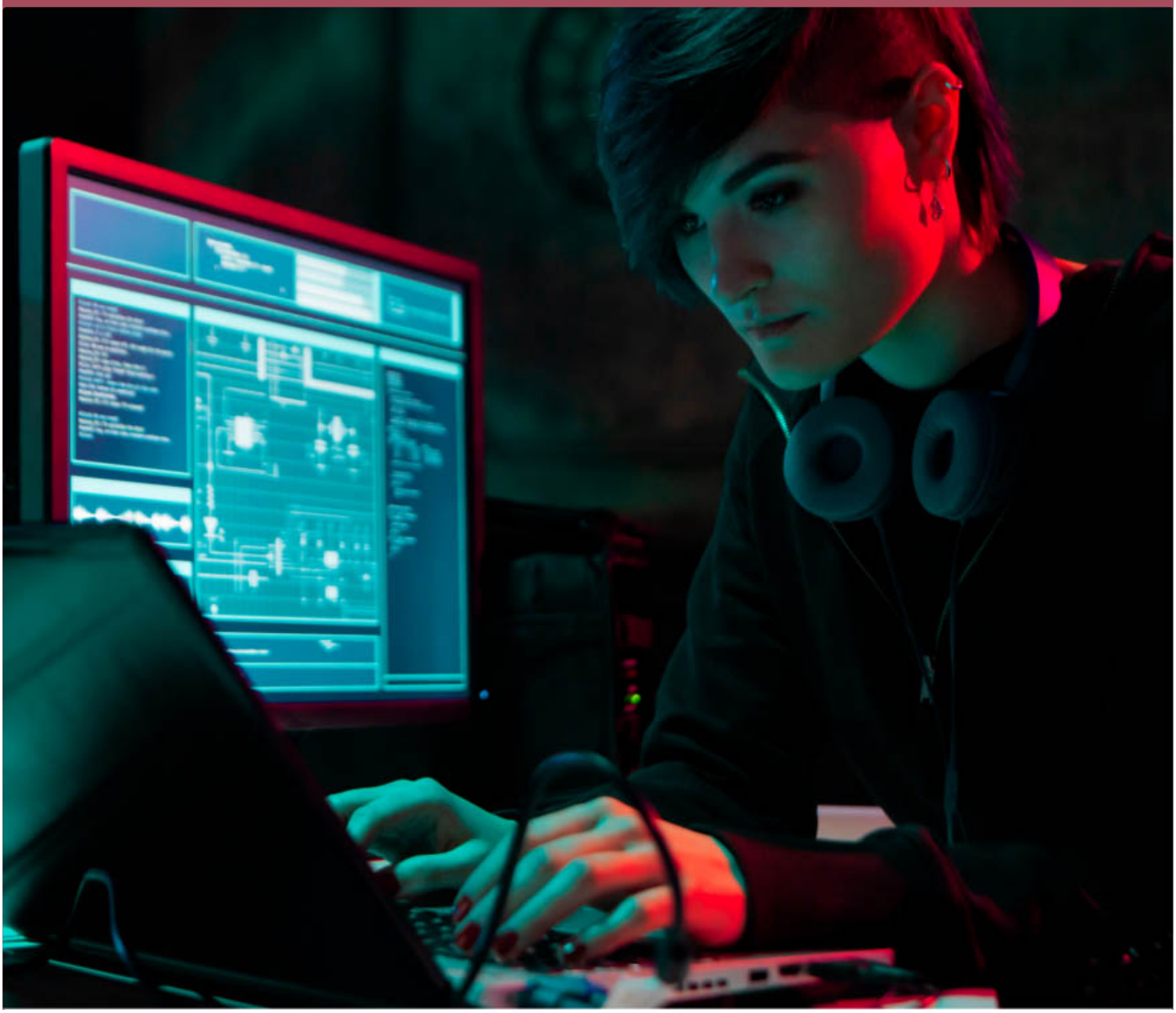
Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

2

Ataques Comuns

Ao final deste módulo, você será capaz de avaliar principais tipos de ataques a instituições e pessoas.



Tipos de vulnerabilidades

Definição

Após a fase de varredura de vulnerabilidades, o atacante possui conhecimento para iniciar ataques direcionados ao seu alvo. Essas vulnerabilidades normalmente são bastante diversas, podendo ir de um serviço de *ftp* configurado sem senha até um serviço de *SMB* desatualizado, permitindo execução de código remotamente após um *buffer overflow*.

Alguns exemplos de vulnerabilidades são erros de configuração de serviços, *Buffer Overflow*, credenciais fracas e vulnerabilidades *zero-day*. Elas podem ser exploradas por atacantes externos e atacantes internos a uma organização ou empresa.

Erros em configuração de serviços

Esse problema ocorre, por exemplo, quando o responsável pela criação ou implementação do serviço o faz de maneira rápida e descuidada, o que pode causar problemas de segurança para o serviço.

O serviço de FTP, por exemplo, possui credenciais padrão conhecidas, em que o usuário é *"anonymous"* sem senha ou com senha *"anonymous"*.

Dessa forma, é possível que um atacante acesse às informações do sistema sem ter que descobrir a senha de outros usuários do serviço.

Buffer Overflow

A vulnerabilidade de *buffer overflow*, ou estouro de *buffer*, permite que, ao enviar determinados dados para uma aplicação, seja possível a manipulação do comportamento do *software* responsável por essa aplicação.

Buffer Overflow na função strcpy() em C

A função *strcpy()*, da linguagem de programação C, recebe dois argumentos, uma *string* com o valor que será copiado, e outra *string* para onde o valor será copiado.

O problema dessa função é que ela não limita quantos caracteres são copiados. Se a *string* de origem for maior que a *string* de destino, a função vai escrever em uma parte da memória depois dos *bytes* reservados para a *string* de destino.

A imagem a seguir mostra um código vulnerável a esse ataque. Na função autenticação, é executada a função *strcpy*, que copia o conteúdo da variável password para a variável *password_buffer*. A variável *flag* tem valor 0, inicialmente. Se a senha estiver correta, o valor é alterado para 1. A senha é testada com a função *strcmp*, que compara as *string*s passadas como parâmetros. Se as duas *string*s forem iguais, o resultado é igual a 0.

Captura de tela de código vulnerável da função autenticação no software vim.

A função autenticação reserva espaço na memória para suas variáveis. A variável *flag*, do tipo *int*, possui 4 *bytes*. A variável *password_buffer* é um array de 7 variáveis do tipo *char*, ou seja, 7 *bytes*.

A função *strcpy* copia os dados do menor endereço para o maior sem limitação da quantidade de caracteres. Se forem enviados vários caracteres 'A', a função *strcpy* irá sobrescrever os valores que estavam armazenados na variável *flag*. A imagem a seguir é apenas uma representação de como o ataque funciona, não sendo fiel ao ambiente real.

Sobrescrita de memória com buffer overflow.

Credenciais fracas

Outro grande problema de segurança é a utilização de **senhas fracas** em sistemas de autenticação. Os atacantes fazem uso de ferramentas capazes de testar muitas senhas em pouco tempo. Quanto mais simples é uma senha, mais rápido é possível descobri-la.

Existem duas formas de executar um ataque em senhas: ataque de *bruteforce*, ou força bruta, e *dictionary attack*, ou ataques de dicionário.

Ataques de dicionário

Os **ataques de dicionário** tentam descobrir a senha do alvo testando uma lista de possíveis senhas. Normalmente, utilizam listas de **senhas fracas** conhecidas como *wordlists*. A mais famosa wordlist é chamada de *rockyou*, que possui as senhas mais utilizadas no mundo como *iloveyou*, *password*, *senha123* etc.

Um cadeado com senha, por exemplo, normalmente possui uma senha com 4 números. Em um ataque de dicionário, seriam testadas senhas comuns como 1234, 1111, 2222 e até mesmo o ano que a pessoa nasceu, como 1985.

Ataques de bruteforce

Em um ataque de *bruteforce*, são testadas todas as possibilidades com dígitos escolhidos. É escolhido um tamanho de senha para testar e serão testadas todas as possibilidades com letras, números e símbolos.

No exemplo do cadeado, um ataque de *bruteforce* seria testar todas as possibilidades de 1111 até 9999.

Vulnerabilidades 0-day ou zero-day

Vulnerabilidades *0-day*, ou dia zero, são vulnerabilidades descobertas por atacantes antes que o desenvolvedor que criou a aplicação tenha tempo de corrigi-las. São chamadas de *0-day* por que o desenvolvedor teve 0 dias para corrigir o problema.

Como essas vulnerabilidades foram descobertas pelos atacantes, existe uma grande probabilidade de o ataque ser bem-sucedido. Elas são uma grande ameaça à segurança, pois o desenvolvedor da aplicação não teve tempo suficiente para corrigir o problema.

Exploração das vulnerabilidades

Exploração de Buffer Overflow

Como já vimos, anteriormente, é possível alterar parte da memória de um programa utilizando a técnica de *buffer overflow*. A exploração dessa vulnerabilidade consiste no envio de código executável malicioso, chamado de *payload*, para a memória e execução desse código manipulando a sequência original do programa, executando o código malicioso.

Os registradores

Os registradores fazem parte do processador de um computador. Seu tamanho, em computadores de 64 *bits* é de 8 *bytes*, ou 64 *bits*. Eles são usados para armazenar informações para execução de cálculos no processador.

O registrador EIP

Um dos registradores mais importantes é o *EIP* (*Extended Instruction Pointer*, ou ponteiro para instruções estendido). Ele é responsável por indicar ao processador qual instrução será executada.

O que são instruções?

As linguagens de programação foram criadas para abstrair o funcionamento do *hardware* que existe em *CPU*. Um computador não entende as linguagens de programação, mas sim a linguagem de máquina.

Um compilador tem a tarefa de transformar o código escrito em alguma linguagem de programação em código de máquina para que o processador seja capaz de executá-la. Esse código de máquina é, basicamente, uma sequência de instruções como soma, subtração, multiplicação, *xor bit a bit, or bit a bit* etc.

Cada uma dessas instruções tem um código, chamado de *opcode*, que é definido em cada processador. A instrução *NOP*, que não faz absolutamente nada, possui o *opcode* 0x90, por exemplo. Essa instrução é utilizada para realizar sincronia entre processos, uma vez que, por mais que não execute nenhuma função em si, demora um tempo para ser executada.

Como alterar o EIP?

Em C, quando uma função é chamada, todo o contexto de execução do programa é alterado. Como mostra a imagem a seguir, o valor de *EIP*, na função principal, em verde, altera seu valor de instrução a instrução, sempre apontando para a próxima instrução a ser executada.

Chamada de função em C e alteração do valor de EIP.

Quando uma função é chamada, o *EIP* tem que alterar seu valor para outro local na memória. Porém, ao final da execução dessa função, ele deve retornar para a instrução 4 da função principal. Para que isso seja possível, o endereço de *EIP* é salvo na memória, da mesma forma que as variáveis de uma função, mas em uma posição diferente.

Então, é possível, com a técnica de *buffer overflow*, alterar o valor de EIP, modificando o comportamento normal do programa.

Valor de EIP alterado com buffer overflow.

Para onde aponta o novo EIP?

Como o atacante é capaz de alterar valores salvos na memória, ele envia, junto com o novo valor de *EIP*, um código malicioso. Além disso, o atacante altera o valor do *EIP*, que antes apontava para o endereço de retorno da função, para um novo endereço com o código malicioso.

Código malicioso na memória utilizada pelo programa em execução.

Na imagem, o *EIP* foi alterado para apontar para o código que o próprio atacante enviou, em vermelho. Em alguns casos, é possível criar acesso através de *Bind Shell* ou *Reverse Shell*.

Bind Shell

A *bindshell* é um tipo de código enviado através de *payload*, que abre uma porta em um servidor.

Envio de bind shell para o servidor.

Com a porta aberta, o atacante se conecta a essa porta e a *bindshell* dá acesso a uma linha de comando no servidor alvo.

Atacante conectado na porta aberta com o bind shell executado no servidor.

Reverse Shell

Reverse shell, ou *shell* reversa, é um código malicioso enviado para o servidor com o objetivo de que ele se conecte ao atacante. Inicialmente, o atacante abre uma porta na sua própria máquina executando um *software* para receber uma linha de comando.

Atacante abre uma porta e envia uma reverse shell para o servidor.

Em seguida, o servidor alvo, ao executar a *reverse shell*, se conecta na porta do atacante e fornece, ao atacante, uma linha de comando.

Servidor se conecta com o atacante via reverse shell na porta aberta.

Ataques de Negação de Serviço

Como ocorrem os ataques de navegação de serviço?

Os **ataques de negação** de serviço procuram explorar vulnerabilidades em protocolos de comunicação para impedir acessos legítimos a determinado sistema. Para a execução do ataque, uma massiva quantidade de pacotes é enviada ao sistema, de forma a sobrecarregá-lo.

DoS – Denial of Service

O *DoS* (*Denial of Service*) é a primeira forma dos ataques de negação de serviço. O ataque é executado a partir de uma única máquina, explorando vulnerabilidades na rede para tornar determinado sistema inacessível.

DDoS – Distributed Denial of Services

Devido ao aumento de proteção em relação aos ataques *DoS*, uma nova forma de ataque foi criada. Para aumentar a quantidade de pacotes enviados para o alvo, os ataques *DDoS* (*Distributed Denial of Service*) utilizam diversas fontes.

Esses ataques são, comumente, realizados a partir de *botnets*, que podem ser definidas como múltiplos dispositivos infectados por um *malware* e controlados por uma central, que envia os comandos do atacante.

Crescimento de ataques DdoS

Existe um crescimento da quantidade de dispositivos que pertencem a alguma *botnet*. De acordo com o relatório de *Threat Intelligence* da *A10 Networks*, 4% de quase 500 mil dispositivos de *IoT* (*Internet of Things*) pertencentes a uma *botnet* estão no Brasil. Ainda de acordo com o mesmo relatório, ataques *DDoS* correm diariamente com diversos alvos.

Entre os motivos que justificam o crescimento de *botnets*, estão o próprio crescimento do número de dispositivos de IoT disponíveis e as novas vulnerabilidades nesses dispositivos.

Ataque SYN Flood

O ataque de negação de serviço do tipo *SYN Flood* (Inundação de *SYN*) tira proveito do funcionamento do TCP e do *3-way handshake* para manter o servidor ocupado com processos desnecessários, indisponibilizando-o.

Basicamente, o atacante envia pacotes *SYN* para o servidor, que responde com um pacote *SYN/ACK*. Porém, o atacante nunca envia um pacote *ACK* para terminar o *3-way handshake*. Devido à natureza do protocolo *TCP*, orientado a conexão, o servidor vai tentar enviar novamente pacotes *SYN/ACK*, até que, depois de um período sem respostas, o processo é encerrado.

SYN Flood, ou Inundação de SYN.

Devido à grande quantidade de pacotes SYN enviados, o servidor se torna indisponível, já que não consegue lidar com todas as tentativas de conexão não estabelecidas.

UDP Reflection

O protocolo *UDP* (*User Datagram Protocol*), diferentemente do protocolo *TCP*, não é orientado a conexão. Ou seja, não existem controles de conexão entre os pacotes e não há garantias que esses pacotes foram enviados de uma máquina para outra. Além disso, não existe a necessidade de nenhum procedimento como o *3-way handshake*. Uma conexão completa com o protocolo *UDP* pode consistir em um cliente

enviando um pacote para um servidor e o servidor respondendo esse pacote.

Funcionamento básico de uma conexão UDP.

Por esse motivo, um ataque de *UDP reflection* pode ser feito enviando pacotes para um servidor que possua algum serviço de *UDP* falsificando o endereço de origem. O servidor envia a resposta desse pacote para o endereço de origem falso, que é o alvo do ataque de negação de serviço.

Ataque de negação de serviço UDP Reflection.

Fator de amplificação

Devido à forma como esses pacotes são enviados no ataque de *UDP Refleciton*, há uma amplificação na quantidade de *bytes* enviados para o alvo. Por exemplo, se um servidor de *DNS (Domain Name System)* receber um pacote de 64 *bytes*, ele pode gerar até 3400 *bytes* de resposta para o alvo.

O fator de amplificação é definido como a razão entre a quantidade de dados enviada para o alvo e a quantidade de dados enviada pelo atacante.

Em um ataque envolvendo o protocolo *DNS*, normalmente o fator de amplificação está entre 28 e 54.

Engenharia Social

O que é Engenharia Social?

Engenharia social é o nome dado ao conjunto de técnicas que tem como objetivo atacar um dos pontos mais fracos de uma organização: o ser humano. Trata-se de uma manipulação psicológica para induzir os usuários a cometerem erros de segurança ou fornecer informações confidenciais.

Para executar essa manipulação, são explorados sentimentos humanos como urgência, medo, empatia, curiosidade, culpa etc. Por isso, normalmente são feitas pesquisas nos alvos para saber como afetá-los emocionalmente.

Exemplo

> Se o alvo for um novo funcionário de uma empresa, pode-se explorar o medo de ser demitido. O ataque pode ser executado informando que o usuário já sofreu um ataque e que é necessário que ele clique em um link para uma página web falsa que irá roubar informações confidenciais do alvo, como senhas utilizadas na empresa.

Esses ataques podem ser executados através de diversos meios de comunicação: telefonemas, *e-mails*, redes sociais, páginas web de compras, pessoalmente etc.

Principais abordagens

Algumas das principais abordagens na execução de ataques de engenharia social são: intimidação, persuasão, bajulação e assistência.

- Na **intimidação**, o atacante assume o papel de uma figura de autoridade, alguém superior hierarquicamente à vítima, para coagi-la a aceitar alguma solicitação. Dessa forma, podem ser explorados os sentimentos de urgência, medo e culpa.
- Na **persuasão**, são utilizadas a lisonja e a menção a nomes de pessoas importantes. A vítima é forçada a seguir as solicitações do atacante pelo sentimento de empatia ou até culpa.
- A **bajulação** é um esquema realizado a longo prazo. O atacante constrói um relacionamento com a vítima para ganhar confiança e, finalmente, informações sobre a vítima.
- Outra possibilidade é que o atacante forneça **assistência à vítima**. Durante a ajuda, o atacante consegue informações confidenciais.

Phishing

Talvez essa seja a **técnica mais conhecida de engenharia social**. Basicamente, consiste em campanhas de *e-mail* e/ou mensagens de texto destinadas às vítimas, com o objetivo de criar um senso de urgência, curiosidade ou medo. Dessa forma, as vítimas eram incitadas a instalar *malware* ou revelar informações confidenciais.

Como são bem genéricos, os ataques de *phishing* podem facilmente ser identificados e bloqueados em políticas de *firewall*.

Spear phishing

O ataque de *Spear Phishing* é uma versão mais direcionada do *Phishing*. O atacante busca atacar pessoas ou empresas específicas, adaptando as mensagens de texto ou e-mails falsos, com base nas características, cargos e contatos das vítimas. Esse ataque requer muito mais esforço do atacante, podendo levar semanas ou meses para preparar e efetuar o ataque.

Ao realizar um ataque de *spear phishing*, o atacante pode, por exemplo, assumir um papel de técnico de TI de uma empresa, solicitando que os destinatários atualizem suas senhas de determinado sistema da empresa. As vítimas vão fornecer as credenciais, pois foram persuadidas a isso.

Como esses ataques são mais específicos, são mais difíceis de detectar.

Os ataques cibernéticos mais comuns

Assista agora a um vídeo em que são apontados os principais ataques cibernéticos mais comuns na internet, com alguns dados para ilustrar.

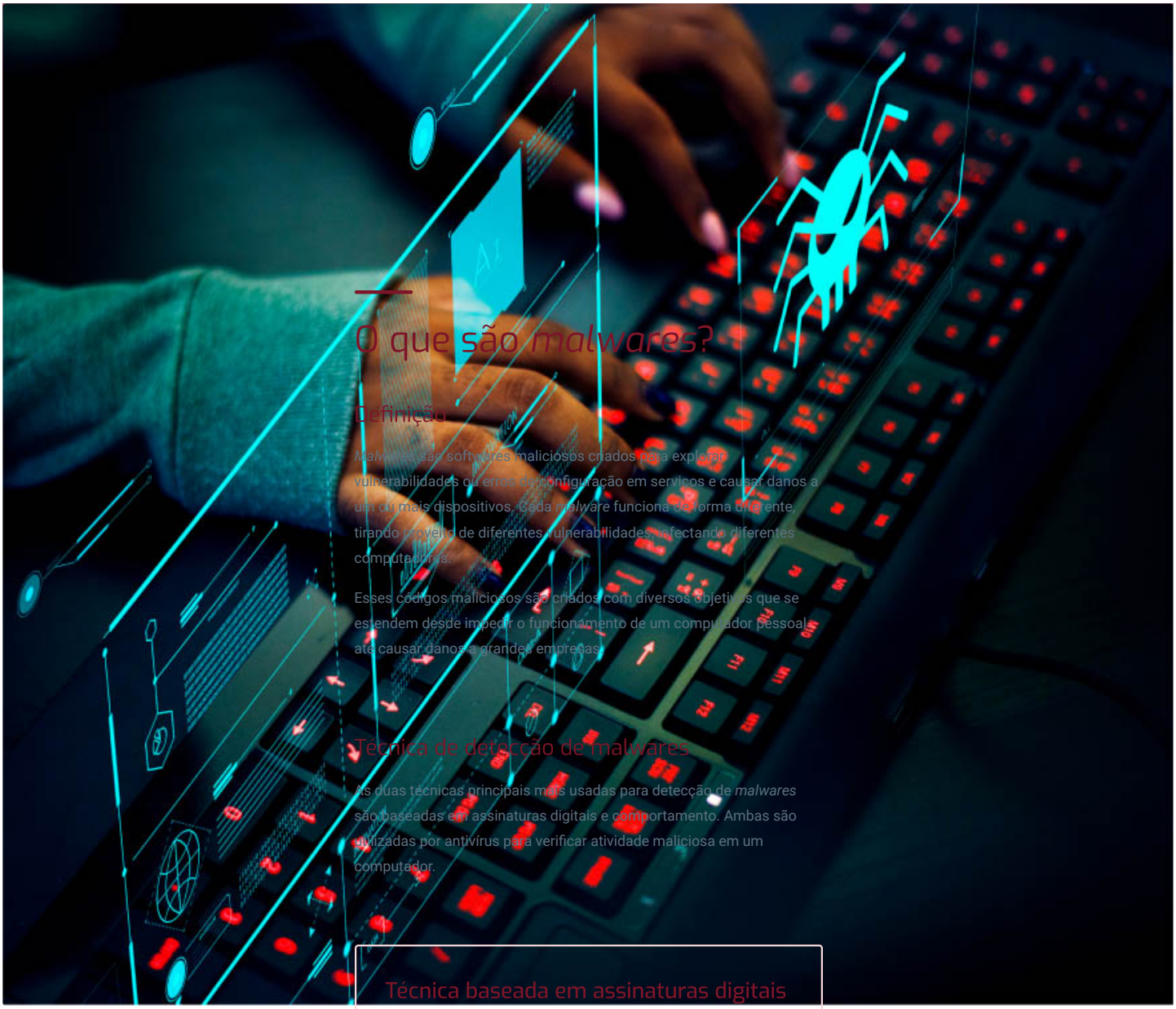
Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

3

Malwares

Ao final deste módulo, você será capaz de classificar os principais tipos de códigos maliciosos utilizados por atacantes.



O que são *malwares*?

Definição

Malwares são softwares maliciosos criados para explorar vulnerabilidades ou erros de configuração em serviços e causar danos a um ou mais dispositivos. Cada *malware* funciona de forma diferente, tirando proveito de diferentes vulnerabilidades, infectando diferentes computadores.

Esses códigos maliciosos são criados com diversos objetivos que se estendem desde impedir o funcionamento de um computador pessoal até causar danos a grandes empresas.

Técnica de detecção de malwares

As duas técnicas principais mais usadas para detecção de *malwares* são baseadas em assinaturas digitais e comportamento. Ambas são utilizadas por antivírus para verificar atividade maliciosa em um computador.

Técnica baseada em assinaturas digitais

Busca por comportamentos específicos que são considerados maliciosos. Ela mantém uma base de dados sobre o que é considerado malicioso. Quando um programa executa alguma dessas atividades, ele é classificado como malicioso.

Técnica baseada em comportamento

A técnica baseada em comportamento, ou anomalia, possui informações sobre o que é considerado como atividade normal. Quando algum programa executa algo fora dos comportamentos normais, ele é considerado malicioso.

Tipos de malwares

Existem diversos tipos de *malware*. Os mais comuns são: *worm* ou vírus, *backdoor*, *botnet*, *downloader*, *launcher*, *Information-Stealing* e *ramsomware*.

Análise estática básica

A análise de *malware* é o conjunto de técnicas utilizadas para examinar algum código malicioso com o objetivo de descobrir informações importantes a respeito de seu comportamento.

Normalmente, essa atividade é executada para realizar uma resposta a um incidente dentro de uma empresa e determinar o que o artefato malicioso pode fazer para, em seguida, detectá-lo com melhor precisão e conter seus danos.

A análise de *malware* pode ser dividida em quatro partes: análise estática básica, análise dinâmica básica, análise estática avançada e análise dinâmica avançada.

Ciclo de uma análise de malware.

As quatro fases, abaixo descritas, são executadas em ciclo, até que sejam exauridas as informações importantes a respeito do objeto a ser analisado.

Exemplos de ataques de malwares

Stuxnet

Stuxnet é um *malware* do tipo *worm* muito famoso por ter atacado uma usina de enriquecimento de urânio. O vírus foi transmitido via dispositivo de armazenamento USB para dentro da usina.

Depois que infectou a primeira máquina, ele começou a se espalhar através do sistema operacional *Windows*, procurando por algum computador que possuísse *PLC* (*Programmable Logic Controller*, ou Controlador Lógico Programável), um controlador e monitor dos equipamentos mecânicos da usina. Quando o encontrou, começou a enviar instruções maliciosas para os equipamentos, danificando-os e interrompendo o correto funcionamento da usina.

É importante ressaltar que a usina não tinha conexão com a Internet e mesmo assim sofreu diversos danos.

Wannacry

Wannacry é um famoso *ransomware* que, em maio de 2017, atingiu mais de 230,000 computadores em todo o mundo. Esse *malware* explora uma vulnerabilidade do sistema operacional Windows, conhecida como *MS17-010*, ou *EternalBlue*, e criptografa todos os dados de uma máquina, solicitando uma quantia em troca dos arquivos.

Essa vulnerabilidade foi descoberta em uma ferramenta, com o mesmo nome *EternalBlue*, da agência *NSA*, *National Security Agency*, em um vazamento de dados executado pelo grupo de hackers chamado *Shadow Brokers*.

Comentário

Foi recomendado por várias empresas que seus empregados não tentassem pagar o resgate por alguns motivos. Primeiramente, os atacantes poderiam estar mentindo em relação à devolução dos arquivos. Além

disso, o pagamento do resgate poderia incentivar futuros ataques, mostrando que as empresas teriam que se redimir ao pagamento. Por último, os atacantes podem ter cometido erros no desenvolvimento do *malware* que tornasse impossível que os arquivos fossem descriptografados.

Mirai Botnet

O *Malware Mirai* é do tipo *worm*, assim como o *Wannacry*, porém, ataca vulnerabilidades em dispositivos de *IoT* (*Internet of Things*, ou Internet das Coisas). Depois de infectados, os dispositivos estão sujeitos a executar comandos recebidos de uma central.

Com isso, é criada uma rede gigantesca de máquinas sob o controle do atacante. É possível então executar ataques de negação de serviço do tipo *DDoS*.

Um dos maiores ataques de negação de serviço relacionados a *Mirai Botnet* é o *DDoS* à *Dyn*, uma das principais empresas de fornecimento de serviços de DNS. Sem a resolução de nomes, vários sites incluindo *Twitter*, *Spotify*, *PayPal*, *Github*, *Reddit* e muitos outros ficaram inacessíveis ao público.

Como se proteger?

O que fazer em caso de ataque?

Diversas são as possibilidades de enviar *malwares* para alvos. Algumas técnicas são simples e outras são complexas. Veremos agora maneiras para você se proteger de ataques.

Visão de geral de tipos de malwares

Assista agora a um vídeo em que são apontados os principais ataques utilizando diferentes tipos de malwares.

Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

4

Wireless Hacking

Ao final deste módulo, você será capaz de distinguir os principais métodos de ataques a redes sem fio.

O que é Wi-Fi?

Definição

As redes sem fio estão presentes em todos os lugares, fornecendo Internet em restaurantes, aeroportos, hospitais etc. Essas redes são comumente chamadas de Wi-Fi, uma abreviação para *Wireless Fidelity* (fidelidade sem fio) e estão formalizadas no padrão IEEE 802.11.

Padrão IEEE 802.11

O Instituto de Engenheiros Eletricistas e Eletrônicos, ou *Institute of Eletrical and Eletronic Engineers*, é uma organização criada nos Estados Unidos com o objetivo de criar padronizações para diversos tipos de tecnologias utilizadas pela humanidade.

Conhecida também por *IEEE*, o instituto criou o padrão 802.11 em 1997. Essa padronização se refere às redes *WLAN*, ou *Wireless Local Area Network*, que em português significa "Rede Local sem fios". A parte 802 é referente à padronização das redes locais e o .11 corresponde especificamente às redes locais sem fio.

Dica

Esses padrões sofrem alterações ao longo dos anos e, para identificar melhor as versões, são adicionadas letras ao final do nome do padrão.

Atualmente, a última versão do 801.11 é "*IEEE Std 802.11axTM-2021*", publicado em fevereiro de 2021.

Conceitos fundamentais

Uma rede Wi-Fi é formada por um *BSS* (*Basic Service Set* ou, em português, Conjunto Básico de Serviço) que pode ser definido como o conjunto de estações Wi-Fi conectadas à rede, em que pelo menos uma delas é um *AP* (*Access Point* ou, em português, Ponto de Acesso).

O *AP*, ou *AP-station*, segundo a nomenclatura definida no padrão 802.11, é o ponto central da rede, em que as demais estações (clientes como notebooks, celulares, equipamentos de IoT etc.) se conectam.

Dica

Cada *BSS* é identificado por um *BSSID*, um número de 48 bits. Normalmente, esse número é igual ao endereço *MAC* da placa de rede do *AP-station*. Um exemplo de *BSSID* é 48-AC-F5-1F-90-5D.

O *SSID* é um nome amigável que podemos dar à rede Wi-Fi. É o nome que encontramos quando o nosso *smartphone* procura por redes.

Faixas de frequência e canais

Outros dois conceitos extremamente importantes são o de faixas de frequência e canais. O WI-FI utiliza faixas de frequência dentro da banda *ISM* (*industrial, scientific and medical*), que é reservada para uso industrial, científico e médico.

A maioria das redes Wi-Fi utiliza duas faixas de frequências. São elas: a faixa de 2,4Ghz e a faixa de 5GHz. Dentro de cada uma dessas faixas, existem os canais, que as separam em pequenas bandas.

A faixa de 2,4GHz é formada por 14 canais de 22MHz cada. Sua extensão completa é de 2.401MHz até 2.495MHz. A tabela a seguir mostra as bandas de frequência e as frequências centrais de cada canal.

Canal	Frequência Central (MHz)	Faixa de frequê (MHz)
1	2.412	2.401 – 2.423
2	2.417	2.406 – 2.428
3	2.422	2.411 – 2.433
4	2.427	2.416 – 2.438
5	2.432	2.421 – 2.443
6	2.437	2.426 – 2.448
7	2.442	2.431 – 2.453

Canal	Frequência Central (MHz)	Faixa de frequê (MHz)
8	2.447	2.436 – 2.458
9	2.452	2.441 – 2.463
10	2.457	2.446 – 2.468
11	2.462	2.451 – 2.473
12	2.467	2.456 – 2.478
13	2.472	2.461 – 2.483
14	2.484	2.473 – 2.495

Tabela - Valores de frequência para os canais 802.11 na faixa 2.4 GHz.
Fonte: Tabela 3.4 de MORENO, 2016, p. 34.

É possível notar que existe superposição entre os canais. Os canais 1, 6 e 11 formam um conjunto de bandas que não se sobrepõem e, por isso, são bastante utilizados. A escolha dos canais é feita na hora de implementação da rede Wi-Fi em algum lugar. Essa configuração pode ser encontrada, de formas diferentes, nos roteadores sem fio.

A faixa 5 GHz possui até 24 canais de 20MHz que não se sobrepõem. A banda utilizada é de 5.150MHz até 5.850MHz. Essa faixa também pode ser dividida em 12 canais de 40MHz cada, dependendo da situação em que será aplicado.

Segurança em Wi-Fi

Redes Wi-Fi OPEN

Quando o Wi-Fi foi difundido inicialmente, não existiam métodos para criptografia e autenticação para essa tecnologia, ou seja, qualquer pessoa poderia se conectar à rede para ter acesso à internet.

O principal problema dessa rede era que qualquer pessoa com um adaptador de rede sem fio poderia capturar os pacotes trocados entre o *Access Point* e o cliente. Como esse tráfego não é criptografado, o atacante poderia ler os pacotes facilmente, sem precisar recorrer a técnicas de quebra de criptografia.

Qualquer pessoa que utilize uma rede Wi-Fi OPEN está sujeita a esse tipo de ataque.

Redes Wi-Fi WEP

Devido à falta de proteção em redes *OPEN*, foi desenvolvida uma forma de criptografia entre o *AP* e o cliente denominada *WEP* (*Wired Equivalency Privacy*, ou privacidade equivalente à rede cabeada).

A ideia do *WEP* é gerar uma chave simétrica para criptografar os dados entre o cliente e o *AP*. Dessa forma, antes de um pacote ser enviado do cliente para o *AP*, e vice-versa, ele é criptografado.

Criptografia simétrica

A criptografia simétrica é definida como um sistema de criptografia em que a chave para criptografar um texto é a mesma utilizada para decodificar a mensagem.

Na rede Wi-Fi, a chave é conhecida pelo cliente e pelo *Access Point* da rede.

Criptografia do WI-FI WEP

Essa criptografia utiliza um algoritmo chamado *RC4* (*Rivest Cipher 4*), baseado na operação *XOR*. Por ser muito simples, essa chave pode ser quebrada com criptoanálise, se muitos pacotes forem capturados.

O ataque se baseia no fato de que alguns pacotes são muito semelhantes, permitindo a análise dos pacotes criptografados para descobrir qual a chave utilizada na criptografia. Por esse motivo, outras formas de criptografia e autenticação foram criadas.

Wi-Fi WEP OPEN

Embora o *WEP* utilize criptografia para proteger os dados, é possível configurá-la sem senha. Nesse caso, dizemos que a rede é *WEP OPEN*. Qualquer pessoa pode ter acesso à rede e, por isso, não é segura contra vários tipos de ataques.

Wi-Fi WEP SKA

A rede *WEP SKA* (*Shared Key Authentication*, ou autenticação com chave compartilhada) adiciona uma camada de proteção. Com isso, é necessário que o usuário conheça uma chave, ou senha, para poder acessar a rede.

WPA

WPA (*Wi-Fi Protected Access*, ou Acesso de Wi-Fi protegido) foi criado com a intenção de ser uma medida temporária para solucionar o problema de segurança em redes *Wi-Fi WEP*, fornecendo novos métodos para criptografia e autenticação.

Criptografia do WPA

O *WPA* utiliza o *TKIP* (*Temporal Key Integrity Protocol*), muito mais robusto que o *RC4*. Nesse método, cada pacote trocado entre o *AP* e o cliente é criptografado com uma chave diferente.

Embora seja mais forte, ainda é baseada no *RC4*, do *WEP*, para realizar a criptografia.

Autenticação do WPA

A autenticação em WPA pode ser realizada de duas formas: *WPA-PSK* (*Pre-Shared Key*, ou Chave compartilhada previamente) ou *WPA-EAP* (*Extensible Authentication Protocol*).

Para a conexão, o cliente troca com o *AP* quatro pacotes. Esse processo é chamado de *4-Way Handshake* ou simplesmente *handshake*. Nessa troca de pacotes, a senha da rede é passada de forma criptografada, mas é possível quebrar essa criptografia utilizando ferramentas.

WPA-PSK

A autenticação com *PSK* provê que o cliente acesse à rede utilizando uma senha para o *Access Point*. Nesse caso, todos utilizam a mesma senha.

WPA-EAP

Diferentemente do *WPA-PSK*, os clientes possuem diferentes senhas para acessar à rede. A autenticação é feita com o *AP* e um servidor *RADIUS* (*Remote Authentication Dial-In User Service*, ou Serviço de Autenticação de Usuário Remoto).

O cliente envia a senha para o *Access Point* que encaminha ao servidor *RADIUS*. O servidor então responde ao *AP* se a conexão foi bem sucedida ou não.

Autenticação WPA-EAP com servidor RADIUS.

Wi-fi WPA 2

Como forma de substituir *WEP* e o *WPA*, foi desenvolvido o *WPA 2*, que utiliza o mesmo método de autenticação do *WPA*, mas usa, como criptografia, o *CCMP* (*Counter Mode Encryption with CBC MAC Protocol*).

O CCMP, em comparação com o *TKIP*, fornece maior segurança, pois utiliza a criptografia *AES* (*Advanced Encryption Standard*), muito mais seguro que o *RC4*.

Ataques a redes sem fio

Tipos de ataques a redes sem fio

Vários tipos de ataques são possíveis em redes sem fio. O principal ataque se refere à quebra de senhas e roubo de informações. O atacante, inicialmente, monitora o tráfego de uma rede e, em seguida, tenta decifrar a senha.

O framework aircrack-ng

O *framework* (conjunto de ferramentas) *aircrack-ng* possui diversos softwares capazes de realizar ataques em redes sem fio. Os principais são *airmon-ng*, *airodump-ng* e *airplay-ng*.

Placas Wi-Fi em modo monitor

As placas Wi-Fi possuem diversos modos de operação. O modo padrão de operação se chama *Manager*, que permite a placa a se conectar em redes sem fio.

É possível alterar o modo de operação de uma placa para o *Monitor*, em que é possível capturar pacotes de diversas redes sem fio próximas, como mostra a imagem a seguir.

Captura de tráfego de uma conexão legítima executada pelo Agente Malicioso.

Pode-se utilizar a ferramenta *airmon-ng*, da suíte *aircrack-ng*, para alterar uma placa para o modo *Monitor*. A imagem a seguir mostra o comando da ferramenta para a ação.

Captura de tela de execução do airmon-ng.

No final da saída do comando, é possível ver o nome *wlp7s0mon*. Esse é o nome definido pelo *airmon-ng* para a interface *wlp7s0*.

Monitoramento de redes sem fio

Com a placa em modo monitor, é possível capturar pacotes de redes sem fio próximas. A suíte *aircrack-ng* possui a ferramenta *airodump-ng*, que mostra informações sobre as redes próximas. A ferramenta pode ser executada como mostra a imagem a seguir.

Captura de tela de comando do airodump-ng.

Na próxima imagem é mostrada uma saída da ferramenta, que fica constantemente atualizando os dados.

Captura de tela de saída da ferramenta airodump-ng.

É possível ver, na saída, o valor de *BSSID*, nome da rede, tipo de criptografia utilizada e o canal das redes *WIFI_SEG* e *Wifi-WPA2*.

Monitoramento de uma rede específica

Por padrão, o *airodump-ng* coleta pacotes e informações de todas as redes próximas. É possível selecionar o canal da rede e o *BSSID* para filtrar a saída da ferramenta, com as opções mostradas na imagem a seguir.

Captura de tela de comando no airodump-ng.

Dessa forma, apenas a rede especificada será monitorada e será possível ver, de forma mais organizada, os clientes que estão utilizando a rede.

Captura de tela de captura de pacotes de rede sem fio com airodump-ng.

Ataques a rede WPA/WPA2

Durante o procedimento de *4-Way handshake*, a senha é trocada entre o AP e o cliente de forma criptografada. O atacante, então, captura essa troca de pacotes para poder tentar quebrar a senha.

Essa captura pode ser feita utilizando a ferramenta *airodump-ng*. Mas, para que o atacante consiga esses pacotes, ele deve forçar o cliente a se desautenticar na rede.

Negação de serviço em redes WPA/WPA2

Nesse ataque, o cliente já está conectado a um *Access Point*. O atacante envia pacotes falsos para o AP para desconectar o cliente. Dessa forma, o cliente vai precisar trocar novamente a senha com o AP e o atacante vai capturar essa troca.

Ataque de negação de serviço e captura do handshake WPA.

O *aireplay-ng*, mais uma ferramenta da suíte *aircrack-ng*, é capaz de forjar esses pacotes com as opções de *BSSID* do *AP*, como mostra a imagem a seguir. Pode ser executada com o seguinte comando.

Captura de tela de execução da negação de serviço com ferramenta aireplay-ng.

Durante todo o procedimento, o atacante utiliza o *airodump-ng* para salvar a troca de dados em um arquivo. Para isso, é necessário executar o seguinte comando.

Execução do airodump-ng para salvar captura em arquivo.

A própria ferramenta mostra quando uma chave é capturada, no canto superior direito sinalizado com *"WPA handshake: D8:77:8B:2B:E4:32"*.

Captura de tela de handshake WPA capturado.

Quebra de senha

O arquivo salvo com o *airodump-ng* possui a senha da rede criptografada. O *aircrack-ng* é capaz de realizar um ataque de dicionário para tentar decifrar a senha. A opção '*-w*' é usada para indicar uma *wordlist* para a ferramenta realizar o ataque.

Captura de tela de comando do aircrack-ng.

Quando a ferramenta encontra a senha quebrada, é mostrada uma tela semelhante à imagem a seguir:

Captura de tela de senha encontrada com aircrack-ng.

É possível ver a senha da rede no meio da tela em *"KEY FOUND! [iloveyou]"*. Nesse caso, a senha é *"iloveyou"*.

Segurança em Redes Wi-Fi

Assista agora a um vídeo em que são apontados os principais ataques cibernéticos em redes sem fios, e as boas práticas utilizadas, para diminuir o risco destes tipos de ataques.

Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

Considerações finais

Como foi visto, os problemas de segurança em sistemas e até em pessoas são diversos. Os atacantes possuem várias ferramentas sofisticadas para elaborar e executar ataques a empresas e pessoas.

Ataques como os mencionados neste conteúdo são recorrentes e fazem parte da realidade do Brasil e do mundo. Cabe aos engenheiros de software, programadores, analistas de rede cuidar da segurança do ambiente de trabalho realizando varreduras em seus sistemas e redes, criando políticas de segurança para o meio digital e corrigindo problemas de segurança ativamente.

Recapitulando os assuntos deste material

Ouça agora um podcast que apresenta uma recapitulação dos assuntos estudados nesse tema, fornecendo ainda exemplos práticos adicionais.