

Capítulo 4 – Vulnerabilidades na Arquitetura TCP/IP

- ✓ 4.1 – Introdução
- ✓ 4.2 – Problemas de segurança inerentes ao TCP/IP
 - 4.2.1 – *Sniffers*
 - 4.2.2 – *Source routing*
 - 4.2.3 – *Spoofing*
 - 4.2.4 – *Syn flood*
 - 4.2.5 – *Smurf*
 - 4.2.6 – *Port scan*
 - 4.2.7 – *Distributed denial of service (DDoS)*

✓ 4.3 – Vulnerabilidades em implementações específicas

- 4.3.1 – Ping da morte (*ping o'death*)
- 4.3.2 – *Teardrop*
- 4.3.3 – *Land*

4.1 - Introdução

- ✓ Neste capítulo, são apresentados alguns ataques conhecidos que são inerentes ao projeto do protocolo TCP/IP
- ✓ Também são apresentados ataques que são problemas de uma implementação específica

4.2 - Problemas de segurança inerentes ao TCP/IP

4.2.1 - *Sniffers*

- ✓ São programas capazes de “escutar” a rede em busca de informações importantes
- ✓ Coleta informações em um determinado segmento de rede
 - comparado a um grampo telefônico, mas permite “escutar” diversas “conversas” ao mesmo tempo
- ✓ A maior parte do tráfego numa rede ainda não é criptografado
- ✓ Exemplos de *sniffers*: Wireshark e tcpdump

- ✓ Podem ser usados para o bem e/ou para o mal
 - bem: análise de tráfego, diagnóstico de problemas
 - mal: coleta de senhas, espionagem
- ✓ *Sniffer* utiliza de um recurso presente nas maiorias das placas ethernet: ***modo promíscuo***
- ✓ Uma NIC geralmente recusa quadros cujo endereço MAC de destino não seja o seu ou um endereço de *broadcast*
- ✓ Em modo promíscuo, a NIC passa a aceitar qualquer quadro, inclusive os que têm destino a outras máquinas da rede
- ✓ ***Hub***: repassa todos os quadros que recebe para todas as portas
- ✓ ***Switch***: só repassa os quadros para as portas certas

- ✓ Uma máquina rodando um *sniffer* não conseguirá ouvir o tráfego presente em outras portas do *switch*
 - ouve somente o tráfego local e os pacotes de *broadcast*
- ✓ Existem técnicas que permitem que o tráfego seja observado mesmo em *switchs*
- ✓ *switchs* possuem *buffer* para armazenar a CAM (tabela de correlações entre endereços MAC e portas do *switch*)
- ✓ *buffer* possui tamanho limitado → CAM terá tamanho fixo com número limitado de entradas

- ✓ Ataque *CAM Table Flooding*: consiste em enviar uma série de requisições ARP falsas, contendo endereços MAC randômicos e distintos
 - essas requisições encherão a tabela CAM do *switch*, fazendo com que novas requisições sejam repassadas para todas as portas
- ✓ Ataque *MAC Spoofing*: falsificação de endereços MAC em quadros ethernet
- ✓ Prevenção: ???
 - configurar manualmente as entradas da tabela CAM para todas as portas do *switch*
 - problemas: estações que se movem com frequência

4.2.2 - *Source routing*

- ✓ *Source routing*: opção especial do pacote IP, em que o originador do pacote pode indicar explicitamente a rota por onde um pacote deve passar
 - opção criada para diagnóstico de rede
- ✓ Usuários maliciosos podem usar o *souce routing* para se fazer passar por outras máquinas na rede
- ✓ Alguns sistemas operacionais já vem com esta opção desabilitada
- ✓ Prevenção: ???
 - configurar todos os roteadores da sua rede para rejeitarem (descartarem) pacotes com essa opção habilitada

4.2.3 - Spoofing

- ✓ *Spoofing* consiste em falsificar identidade na rede
 - é qualquer procedimento que envolva personificação de usuários ou máquinas, incluindo endereços IP e consultas em servidores de nomes
- ✓ Usado para obter acesso não autorizado, ou para esconder tentativas de ataque
- ✓ Existem diversos tipos de *spoofing*:
 - IP
 - DNS
 - Web
 - e-mail
 - roteamento
 - etc

E-mail spoofing

- ✓ O SMTP não possui nenhum sistema de autenticação padrão que comprove a autenticidade do remetente
- ✓ Forma simples de realizar um ataque de engenharia social
- ✓ Forma mais simples e utilizada para realizar um *spoofing*
- ✓ Como fazer ???
 - basta informar ao SMTP, no momento do envio do e-mail, um remetente diferente
 - SMTP não confere a identidade do remetente, assim como no correio convencional :(
- ✓ Como evitar ???
 - Configurar o MTA para que ele exija autenticação
- ✓ Recomendações sobre e-mail *spoofing*:
 - http://www.cert.org/tech_tips/email_spoofing.html

IP spoofing

- ✓ *IP spoofing* consiste em falsificar o endereço IP de origem para tentar enganar regras de filtragem por IP, ou sistemas de autenticação baseados em IP
- ✓ O roteamento na Internet se baseia no IP de destino
- ✓ O IP de origem é usado apenas para o retorno da informação
- ✓ Pacote com IP de origem falso chegará ao destino, mas sua resposta será enviada para outra origem
- ✓ Parece uma prática inútil, uma vez que o pacote nunca retornará para o próprio atacante
- ✓ Por que fazer ???

- ✓ Sequestro de conexão: em conexões em andamento, o atacante pode conseguir “sequestrar” a conexão, através da manipulação dos números de sequência do TCP
 - utilizar um *sniffer* e capturar as respostas
 - necessário que o atacante esteja na mesma sub-rede que a vítima
- ✓ Ataque cego: o atacante é capaz de “advinhar” qual seria a resposta e envia o próximo pacote de acordo
 - necessário “advinhar” os números de sequência do servidor
- ✓ Negação de serviço (DoS): o atacante enviará pacotes mal-formados com a intenção de negar o serviço remoto através de uma vulnerabilidade ou inundação

- ✓ Como se proteger ???
- ✓ Usar filtros ***ingress*** e ***egress***: nenhum pacote deve sair de sua rede com um endereço IP que não pertença à sua rede; e nenhum pacote deve entrar com endereço de origem de dentro da sua rede
- ✓ Números de sequência randômicos
- ✓ Criptografia e autenticação: sistemas seguros devem criptografar a informação e exigir meios de autenticação entre as partes

Roteamento

- ✓ **Roteamento estático:** definido manualmente pelo administrador da rede
- ✓ **Roteamento dinâmico:** definido através de protocolos específicos de roteamento
- ✓ Rotas dinâmicas são propagadas pelos próprios roteadores que compõem a rede
- ✓ Atualizações de tabelas de roteamento podem ser falsificadas
- ✓ Como fazer ???
 - qualquer elemento na rede poderia se passar por um roteador e enviar atualizações de rota falsas para outros roteadores da rede
- ✓ Como evitar ???
 - utilizar protocolos modernos, como o RIPv2 e o OSPF, que dispõem de esquemas de autenticação

4.2.4 - Syn flood

- ✓ TCP usa *three way handshake* para estabelecer conexão
 - 3 pacotes são necessários para estabelecer uma conexão TCP:
 - 1 - pedido
 - 2 - resposta com um pedido
 - 3 - resposta final
- ✓ Num ataque *syn flood*, o atacante envia uma série de pacotes com o *flag* SYN ativado
 - normalmente com endereço IP de origem falsificado
 - confirmação final nunca chega
- ✓ *Buffer* utilizado para armazenar conexões pedidas e não efetivadas é limitado
 - quando encher, o servidor deixará de aceitar novos pedidos de conexão até que o *buffer* seja esvaziado

- ✓ Um pedido possui *timeout*, com o tempo o *buffer* se esvaziará sozinho
- ✓ Muito utilizados para retirar servidores de funcionamento temporariamente, impedindo que o servidor legítimo responda aos clientes
 - nesse momento, o atacante poderá enviar respostas falsas para os clientes que estão pedindo conexão: spoofing
- ✓ Não existe uma forma 100 % de impedi-lo
- ✓ Como amenizar?
 - limitar o número de pacotes SYN por segundo através de um *firewall*
 - Efeitos colaterais?

4.2.5 - Smurf

- ✓ São ataques que utilizam um recurso do TCP/IP chamado *directed broadcast*
 - endereço de *broadcast* permite o envio de um pacote para todas as máquinas de determinada rede
 - pacotes *broadcast* geralmente ocorrem dentro de uma mesma rede
- ✓ Quando uma máquina de uma rede envia um *broadcast* para uma outra rede => *broadcast* direcionado
- ✓ Ataque *smurf*:
 - 1 - o atacante procura uma série de redes com um grande número de máquina em cada uma
 - 2 - o atacante obtém o endereço de *broadcast* de cada uma dessas redes => chamadas de redes amplificadoras

- 3 - o atacante envia uma série de pacotes com o endereço IP de origem falsificado (IP *spoofing*) igual ao endereço IP da vítima para os *broadcasts* das redes amplificadoras
- 4 - o pacote enviado pode ser simplesmente do tipo ICMP *echo request (ping)*
- 5 - as máquinas das redes amplificadoras responderão com um *echo reply* endereçados ao IP da vítima
- 6 - a vítima será inundada com pacotes ICMP de resposta
- 7 - a quantidade excessiva de respostas causará um DoS na vítima

✓ Não existe uma forma eficiente para a vítima se proteger

- redes amplificadoras poderiam bloquear pacotes *directed broadcast*
- máquinas devem ser configuradas para não responderem ao *echo request (ping)*

4.2.6 - Port scan

- ✓ Não é considerado um ataque em si => apenas uma atividade de reconhecimento
- ✓ Consiste em tentar se conectar em todas as portas de uma máquina, na tentativa de descobrir que serviços estão ativos naquela máquina
- ✓ Um *port scan* normalmente é a primeira medida que um atacante tomará em relação a seu servidor
 - servirá para o atacante saber quais serviços estão ativos para depois pesquisar vulnerabilidades nesses serviços
- ✓ Num *port scan*, a informação de retorno é muito importante para o atacante, geralmente não utilizam IP's falsos => “fácil” descobrir a origem do ataque?
- ✓ Um *port scan* pode ser detectado através de um IDS ou dos *logs* do sistema operacional

4.2.7 - Distributed denial of service (DDoS)

- ✓ Mesmo que o DoS, só que realizado de forma coordenada, no qual muitas origens estão envolvidas
- ✓ São difíceis de executar, mas muitos ataques deste tipo já ocorreram em larga escala, deixando sites indisponíveis por horas
 - Yahoo!
 - eBay
 - Microsoft Update
- ✓ Ataque DDoS:
 - 1 - o atacante invade um grande número de máquinas na Internet
 - 2 - o atacante instala um programa DDoS nas máquinas invadidas. Esse programa permite ao atacante controlar remotamente a máquina, transformando-a num “zumbi”

- 3 - a partir de um controle central, o atacante pode direcionar todos os seus “zumbis” para atacar uma vítima, realizando cada uma um ataque DoS

- ✓ Ataques DDoS utilizam ataques *smurf*, pacotes fragmentados ou requisições de serviço
- ✓ Objetivo é fazer com que o servidor processe muito mais informação do que ele seria capaz de processar
- ✓ O ataque DDoS é muito difícil de ser rastreado, pois geralmente utilizam IP de origem falsificado, além de utilizar “zumbis”
- ✓ Atividade relacionada ao DDoS que tem crescido bastante: *botnets*
 - são redes de máquinas infectadas com algum programa malicioso que faz com que um atacante seja capaz de ter controle total sobre essas máquinas
 - e.g.: máquinas que se conectam automaticamente a um servidor IRC comprometido. O atacante envia comandos via servidor IRC a todas as máquinas conectadas a ele

4.3 - Vulnerabilidades em implementações específicas

4.3.1 - Ping da morte

- ✓ Utilitário *ping* trabalha com pacotes ICMP *echo request* e *echo reply*
- ✓ Apesar do utilitário dispor de uma forma de alterar o tamanho do pacote desejado, este não excedia 64 Kbytes
- ✓ Em 1996, o programador do utilitário se “esqueceu” da limitação do tamanho na versão do Windows
 - o *ping* do Windows era capaz de enviar pacotes maiores que 64 Kbytes

- ✓ As implementações TCP/IP de muitos servidores, roteadores, impressoras, etc, não foram preparadas para suportar um pacote *ping* maior que 64 Kbytes
- ✓ Também não possuíam mecanismos de checagem do tamanho do pacote *ping* antes de o copiarem para a memória
- ✓ Um simples “*ping*” era capaz de fazer com que diversos equipamentos travassem ou reiniciassem
- ✓ Praticamente não existem mais sistemas vulneráveis ao *ping* da morte

4.3.2 - Teardrop

- ✓ Explorava uma vulnerabilidade em implementações do IP que não faziam verificações no campo *fragmentation offset*
- ✓ O ataque enviava pacotes com valores negativos ou sobrepostos
- ✓ Causava DoS
- ✓ SO's afetado: Windows NT 4.0, Windows 95 e Linux

4.3.3 - Land

- ✓ Consistia em enviar pacotes SYN com a porta e o IP de origem iguais a porta e o IP de destino
- ✓ Fazia com que o Windows NT 4 sofresse de excesso de processamento e o Windows 95 travasse

4.4 - Atividades

1. Usar o tcpdump no roteador de sua bancada para capturar todo o tráfego da rede interna.
2. Analisar o tráfego capturado na atividade 1 dentro do Wireshark.
3. Utilizar o Wireshark para analisar o tráfego interno da rede usando um hub e depois um switch.
4. Configurar, manualmente, a placa de rede de uma estação para operar em modo promíscuo
5. Pesquisar na Internet uma forma de se fazer um ataque CAM Table Flooding no switch de sua bancada. Comprovar o sucesso desse ataque com o Wireshark.
6. Realizar um ataque MAC Spoofing e comprovar o sucesso desse ataque utilizando o tcpdump no roteador de sua bancada.