Segurança de redes de computadores

Roberto Sadao Yokoyama

UTFPR-CP

Setembro, 2016

Aula de hoje: [1]

[1] Slides baseados no material do livro: Nakamura [3]

Dumpster diving ou trashing

- Atividade na qual o lixo é verificado em busca de informações sobre a organização ou rede da vítima
- Lista telefônica corporativa, organograma, memorandos internos, manuais de política, calendário de reuniões, manuais de sistemas, impressão de código-fonte
- Esta técnica é eficiente e muito utilizada
- É necessário a organização ter uma política de descarte de lixo

Dumpster diving ou trashing

"(...) Casos de espionagem ilegal são mais comuns quanto maior é a rivalidade entre os envolvidos. Veja o caso da americana Procter & Gamble, uma das maiores empresas de produtos de higiene e limpeza do mundo e concorrente histórica da anglo-holandesa Unilever. Em 2001, a P&G contratou uma empresa de investigação para tentar descobrir detalhes sobre o projeto de um novo xampu que vinha sendo desenvolvido pela Unilever. Num lance desastrado, **um dos espiões foi pego revirando o lixo da Unilever atrás de informações secretas**. Como consequência, a P&G foi obrigada a fazer um acordo e pagar à concorrente 10 milhões de dólares em indenizações (...)[2]"

[2] http://exame.abril.com.br/revista-exame/edicoes/823/noticias/o-rastro-da-espionagem-m0041505

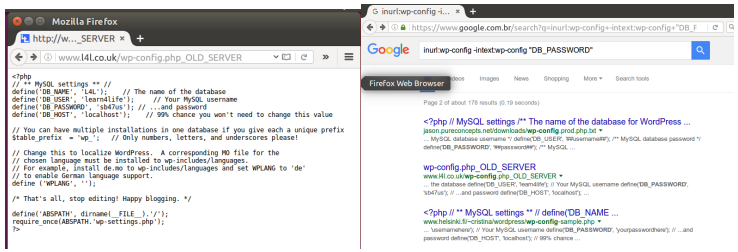
Ataque de obtenção de informações livres

- As diversas informações que podem ser obtidas livremente, principalmente na própria rede:
  - Hosts ativos (ping)
  - IP do gateway (route)
  - Endereços da sub-rede (ifconfig)
  - DNS (nslookup)
  - Roteadores (traceroute)
- Análise de cabeçalhos de e-mail e busca de informações em listas de discussão
- Mecanismos de busca. Ex.: Google
- Redes sociais. Ex.: LinkedIn, onde há informações sobre cargos e funções de usuários
- São considerados métodos não intrusivos

Ataque de obtenção de informações livres

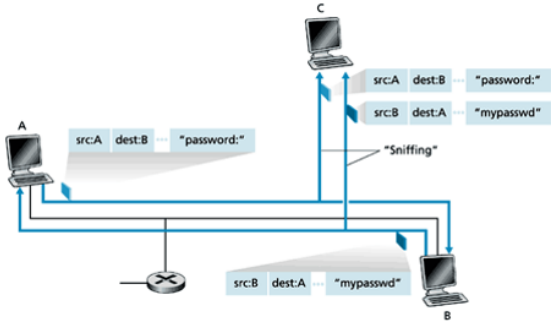
Google Hacking

Exemplo: inurl:wp-config -intext:wp-config "DB_PASSWORD"



Packet sniffing

- Também conhecida como *Passive Eavesdropping*, essa técnica consiste na captura de informações diretamente pelo fluxo de pacotes na rede
- Softwares: TCPDump, Wireshark, WinDump etc
- As informações que podem ser capturadas pelos *sniffers* são referentes aos pacotes que trafegam no mesmo seguimento de rede
- E-mails, senhas (FTP, telnet), etc, que trafegam abertamente pela rede podem ser facilmente capturadas dessa maneira



Packet sniffing

Detecção de sniffer:

- **MAC detection**: quando o SO não confere o endereço MAC em modo promiscuo. A técnica utiliza uma mensagem ICMP com MAC falso e endereço de IP do host a ser verificado. Se tiver resposta é porque host pode estar executando um sniffer.
- **Inspeção**: administrador acessa a máquina e verifica se existe um sniffer em execução e/ou interfaces em modo promiscuo
- **Utilização de "armadilhas"**: tráfego de senha "isca" (transmitido em texto claro) e verificar se a senha foi utilizada

Packet sniffing

Detecção de sniffer: (continuação)

- **DNS detection**: alguns sniffer realizam DNS reverso. Tráfego com endereços falsos são colocados na rede, caso o sniffer capture esses pacotes, o pedido de DNS reverso será enviado ao servidor. Essa técnica pode ainda detectar sniffers em diferentes seguimentos de rede.
- **Load detection**: a ideia dessa técnica é que os equipamentos que estão executando sniffers têm maior grau de processamento, e assim levam mais tempo para responder as requisições. Essa técnica não funciona em redes com grande trafego. Não pode ser utilizados pacotes como ICMP, deve ser um método no nível de usuário, como comandos FTP.

Packet sniffing

Algumas técnicas para driblar os switches:

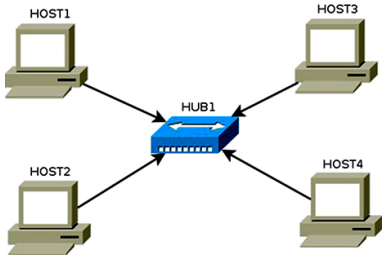
- Acesso administrativo ao equipamento
- Reconfiguração do switch via uso de *Simple Network Management Protocol* (SNMP)
- Encher a tabela MAC do switch enviando diversos quadros com endereços MAC novos
- Envio de quadros com endereço ARP (*Address Resolution Protocol*) falsos, fazendo com que o tráfego de outro equipamento seja enviado para o equipamento do atacante.

Algumas medidas de segurança:

- Segmentar a rede: switches (VLANs) e roteadores
- Utilizar lista de controle de acesso baseado em MAC
- Tabela MAC estática
- Encriptar os dados

Packet sniffing

Netkit: Lab01 - Uma rede simples conectada por um hub[3]



- Experimentar configurações de rede sobre um domínio de colisão simples
- Estudar a composição dos pacotes, desmontando os no wireshark

[3] Gurguel [1]

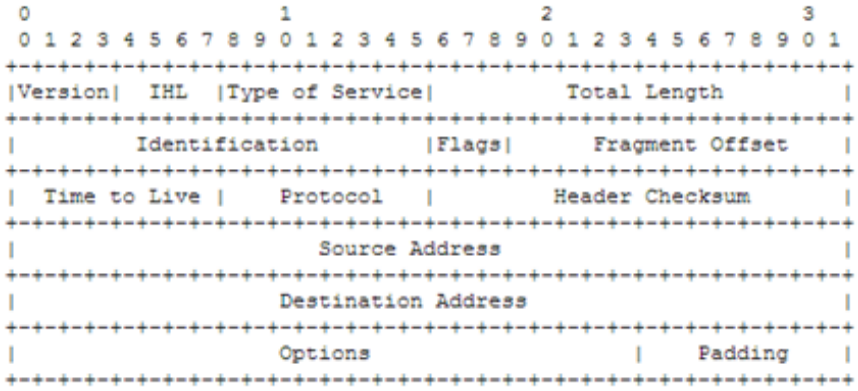
Packet sniffing

Testando com serviço FTP

1. Crie um usuário em HOST1: *adduser usuario*
2. Inicie o servidor FTP: */etc/init.d/proftd start*
3. Inicie o TCPDUMP no HOST 3 ou HOST 4: *tcpdump -i eth0 -v -n -w lab1_hostX.pcap*
4. Acesse o servidor FTP via HOST2: *ftp 192.168.1.1*
5. Pare a captura de pacotes: *CTRL+C*
6. Analise os pacotes no wireshark

Packet sniffing

Análise do protocolo IPv4[4]



https://www.sans.org/security-resources/tcpip.pdf

[4] Mota Filho [2]

Packet sniffing

Análise do protocolo IPv4

- **Version**: 4 bits (1/2 byte), normalmente 0100
- **IHL**: Internet Header Length, 4 bits, informa quantas linhas há no cabeçalho (cada linha 4bytes) – mínimo 5 (20 bytes) e máximo 15 (60 bytes)
- **ToS**: Type of Service, 8 bits, determina a QoS. Hoje a maioria dos host e roteadores ignora o conteúdo ToS
- **Total Length**: tamanho total do pacote IP (cabeçalho+payload) (máximo 65535). Para calcular o tamanho do payload: $IHLx4 - TotalLength$

Packet sniffing

Análise do protocolo IPv4

- **Identification**: trata da fragmentação dos pacotes. Um número identifica o pacote IP (16bits), gerado a cada conexão de maneira aleatória. Todavia pacotes de uma mesma conexão terão números seriais.
- **Flags**: ajuda a controlar o fluxo de fragmentos IP, quando ocorre fragmentação. Possui 3 bits (none, DF e MF). None – uso futuro. DF – don't fragment (1 não fragmenta), MF – more fragment (1 mais fragmentos)
- **Fragment Offset**: refere-se ao byte inicial de cada fragmento divido por 8 (13 bits). O cálculo considera somente o payload. Ex. 3000 bytes: 0/8=0; 1480/8=185 e 2960/8=370

Packet sniffing

Análise do protocolo IPv4

- Para testar: ping -c 1 -s 3500 192.168.1.1
- tcpdump -n icmp and host 192.168.1.2 -v

Packet sniffing

Análise do protocolo IPv4

- **TTL**: quantidade de roteadores transpostos, que o pacote pode trafegar (8 bits). Cada roteador ultrapassado decrementa 1. Ex. Windows 128 e Linux 64
- **Protocol**: diz o que será "encontrado" no payload do IP (ICMP, TCP, e UDP) (8 bits). Ex. ICMP=1; TCP=6 e UDP=17.
- **Checksum**: verificar a integridade do cabeçalho (16 bits). Qualquer alteração no cabeçalho irá alterar o checksum.

Packet sniffing

Análise do protocolo IPv4

- **Source Address**: endereço IP de origem (32 bits).
- **Destination Address**: endereço IP de destino (32 bits).
- **Options**: não é obrigatório. O tamanho pode variar de 0 a 40 bytes. Ex. timestamp; Strict source routing – lista de roteadores que o roteador pode passar
- **Padding**: só existirá se houver Options. Trata-se bits completadores.
- **Payload**: área de dados.

Onde está a máscara de rede?

Packet sniffing

Implementando um sniffer[5]

- Normalmente é necessária uma biblioteca chamada PCAP
- Para capturar pacotes vamos utilizar um tipo especial de socket chamado de socket puro (raw socket)

```
class IP(Structure):
    _fields_ = [
        ("ihl",             c_ubyte, 4),
        ("version",         c_ubyte, 4),
        ("tos",             c_ubyte),
        ("len",             c_ushort),
        ("id",              c_ushort),
        ("offset",          c_ushort),
        ("ttl",             c_ubyte),
        ("protocol_num",    c_ubyte),
        ("sum",             c_ushort),
        ("src",             c_ulong),
        ("dst",             c_ulong)
    ]
```

[5] Seitz [4]

Atividade

1) Implemento/Modifique um *sniffer* de rede para exibir as seguintes informações sobre os pacotes capturados:

- Versão
- IP de origem
- IP de destino
- Protocolo encapsulado no payload
- Tamanho do cabeçalho e do payload
- Valor do TTL
- Fragmentação (identificação, se foi fragmentado e o valor do offset)

Referências

[1] P. Gurgel, K. R. L. C. Branco, L. H. C. Branco, F. E. Barbosa, and M. M. Teixeira. *Redes de Computadores Da teoria à prática com Netkit*. Elsevier, 1ed, 2015.

[2] J. E. Mota Filho. *Análise de tráfego em redes TCP/IP*. Novatec, 2013.

[3] E. T. Nakamura and P. L. Geus. *Segurança de redes em ambientes cooperativos*. Novatec, 2007.

[4] J. Seitz. *Black Hat Python – Programação Python para hackers e pentesters*. Novatec, 2015.