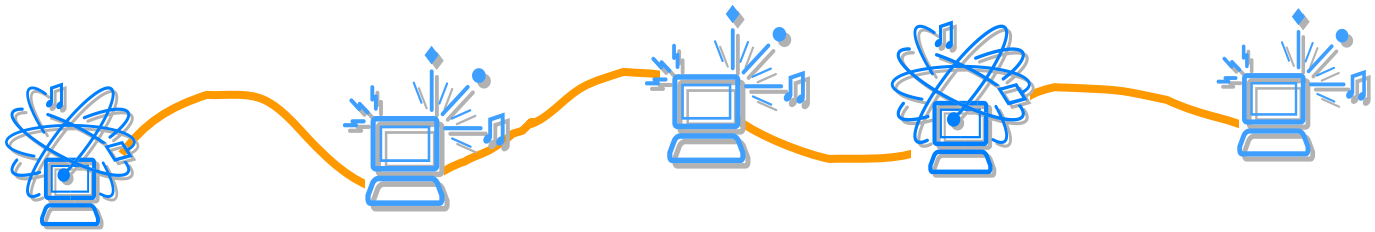
Pensando como um *hacker*



Baseado em:
Thinking like a hacker
Por Eric Schultze, Chief Security Architect, Shavlik Technologies

Pensar como um *hacker*

- Pensar como um *hacker* não é muito diferente de pensar como um bom analista/programador
- Aplicam paciência e documentam cada passo do seu trabalho
- O objectivo dum *hacker* é comprometer um determinado sistema alvo ou aplicação
- O *hacker* começa normalmente com pouca ou nenhuma informação sobre o alvo; quando termina a análise o atacante construiu um mapa detalhado do caminho que lhe permitirá comprometer o alvo
- Só pode atingir os objectivos através da análise cuidadosa e aproximação metódica à futura vítima

Método sistemático em 7 passos

1. Efectua uma análise das “pegadas”
2. Enumera a informação
3. Obtém acesso através da manipulação dos utilizadores
4. Escala os privilégios
5. Recolhe *passwords* e segredos adicionais
6. Instala *backdoors*
7. Tenta controlar outros sistemas a partir do sistema comprometido

Análise das “pegadas”

- Identifica os vários domínios de nomes que interessa investigar
- Recolhe toda a informação através de recursos públicos
- Esta análise permite ter uma indicação da dimensão do alvo, quantos potenciais pontos de entrada podem existir e quais os mecanismos de defesa que poderão existir para retaliar o ataque

Informação útil para o ataque

- Nomes da companhia
- Nomes de domínios
- Parceiros de negócio – subsidiárias
- Redes IP
- Números de telefone

Detecção dos pontos fracos

- Em vez de tentar furar através dos *firewalls* principais da empresa, “armados até aos dentes”, os *hackers* procuram outros pontos fracos como subsidiárias, filiais, etc. que possam fornecer acesso à rede da empresa, talvez através de VPNs, sem passar as defesas principais

Ferramentas mais usuais

- Os ***port scanners*** são utilizados para determinar quais as máquinas que são acessíveis, quais os portos UDP e TCP activos em cada sistema e o sistema operativo existente em cada máquina.
- São efectuados ***traceroutes*** para ajudar a identificar a relação de cada máquina com cada uma das outras e identificar potenciais mecanismos de segurança entre o atacante e o alvo

Mapear a rede

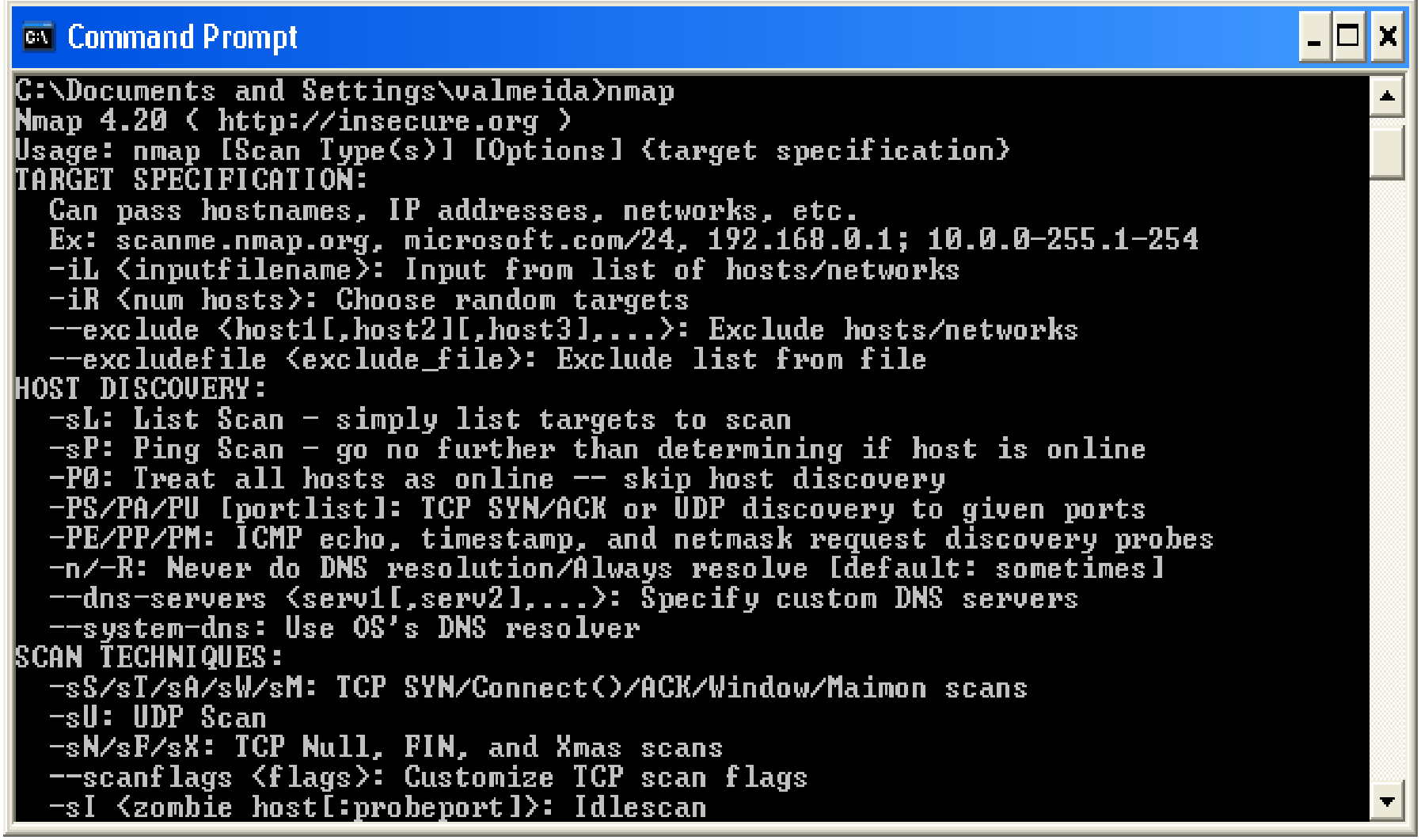
- Depois da análise da rede o atacante pode criar o mapa daquela.
- O mapa é utilizado para a próxima fase no ataque:
  - Enumeração da informação

Ferramentas

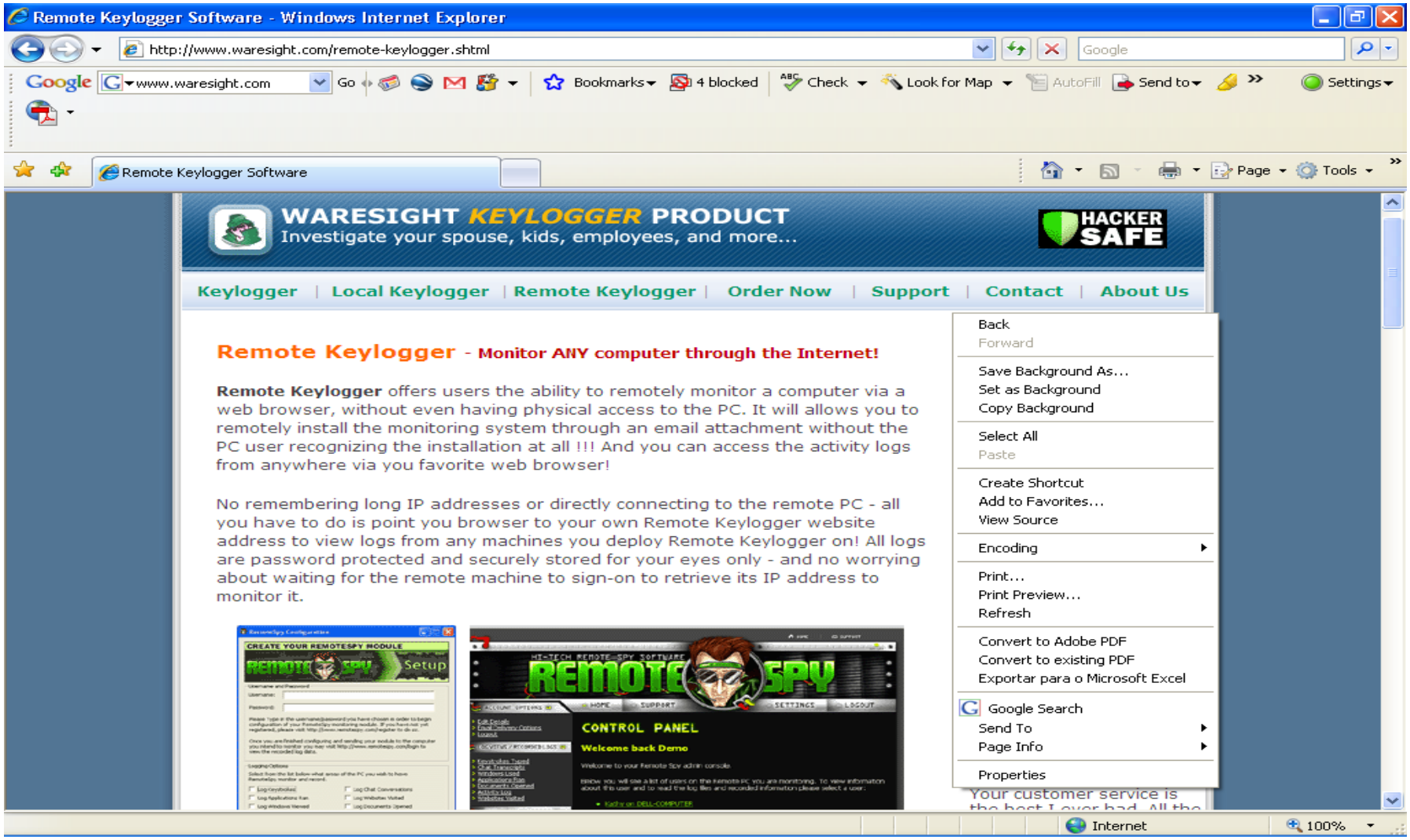
Exemplos

- **nslookup** – *Queries* de DNS [Windows]
- **tracert** – Realização de ***traceroutes*** [Windows]
- **Nmap** – *Scanner* de portas [http://insecure.org/nmap/]
- **McAfee Free Tools** – Dezenas de ferramenta relacionadas com segurança [http://www.mcafee.com/us/downloads/free-tools/index.aspx]

NMAP


Command Prompt
C:\Documents and Settings\valmeida>nmap
Nmap 4.20 (http://insecure.org)
Usage: nmap [Scan Type(s)] [Options] <target specification>
TARGET SPECIFICATION:
Can pass hostnames, IP addresses, networks, etc.
Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
-iL <inputfilename>: Input from list of hosts/networks
-iR <num hosts>: Choose random targets
--exclude <host1[,host2][,host3],...>: Exclude hosts/networks
--excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
-sL: List Scan - simply list targets to scan
-sP: Ping Scan - go no further than determining if host is online
-P0: Treat all hosts as online -- skip host discovery
-PS/PA/PU [portlist]: TCP SYN/ACK or UDP discovery to given ports
-PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
-n/-R: Never do DNS resolution/Always resolve [default: sometimes]
--dns-servers <serv1[,serv2],...>: Specify custom DNS servers
--system-dns: Use OS's DNS resolver
SCAN TECHNIQUES:
-sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
-sU: UDP Scan
-sN/sF/sX: TCP Null, FIN, and Xmas scans
--scanflags <flags>: Customize TCP scan flags
-sI <zombie host[:probeport]>: Idlescan

Desenvolvimento não necessário



Desenvolvimento não necessário

Pontos a considerar

- Qual o rasto que a aplicação utilizada deixa no sistema operativo?
- Pode-se confiar na aplicação ou, se estiver comprometida, possibilitará o acesso à sua máquina?
- Qual a informação que a aplicação ou sistema apresenta aos utilizadores não autenticados?
- Que portas abre o software no sistema? Pacotes mal formados ou ataques de *flooding* pararão o serviço, consumirão memória ou ciclos de CPU?
- Existem *firewalls*, ou protecções na aplicação que evitem um ataque pela “porta da frente”?

Enumerar informação

- Após os *hackers* terem efectuado a análise das pegadas e gerarem o mapa que aumenta o seu conhecimento sobre o sistema alvo, tentam obter tantos dados quanto possível do sistema alvo.

Técnicas de ataque

- *Eavesdropping*
  - *Sniffing da rede*
    - Fácil de realizar em LAN com *broadcast* se o atacante conseguir ter acesso a uma ligação física à rede
    - As Wireless LAN usam por vezes métodos de cifra inadequados
    - A segurança nas WAN varia
  - *Keystroke logging*
    - O software para realizar *keystroke logging* pode ser instalado por vírus ou directamente pelo atacante
    - Alternativamente, pode ser inserido hardware para captura

Técnicas de ataque

- Ataques de autenticação
  - *Crack* de *passwords*
    - Tentar todas as possibilidades
    - Sniffing da rede
    - Velho método de “shoulder-surfing” (espreitar por cima do ombro
    - Também, *key loggers*, virus, etc, como referido
- Software malicioso
  - Código “móvel”
  - Oportunidades para:
    - Virus, *worms*, cavalos de Tróia (*Trojan horses*), *spyware*, …

Técnicas de ataque

- Engenharia social
  - Pratica para obter informação confidencial através da manipulação de utilizadores legitimos (normalmente através de conversa “fofa”)
  - Os utilizadores são levados a fornecer *passwords* ou outros segredos ou a permitir aos atacantes ultrapassar a segurança
  - Muto comum (e eficaz)
    - Tipicamente “atacam-se” um grande grupo de utilizadores na esperança de encontrar o “elo mais fraco”, se este não for evidente logo ao inicio
    - Na maioria das vezes usado de forma muito subtil

Técnicas de ataque

- Engenharia social – alguns exemplos
  - Em Março de 2005, nos EUA, mais de um terço dos funcionários, incluindo gestores, das finanças que foram contactados por inspectores do departamento do tesouro do governo, a fazerem-se passar por tecnicos da informática, divulgaram os seus logins e passords.
  - A ameaça designada de virús designada por "Teddy Bear" levou a que muitos utilizadores assustados apagassem o "jdbgmgr.exe" (critico para alguns utilizadores). jdbgmgr.exe tem um icom com um "teddy bear", o qual pareceu suspeito a muita gente.
  - Em Março de 2005 muitos utilizadores receberam um email pressupostamente enviado pelo banco da Irlanda. Esse email possuia um *link* para um *site* falso muito semelhante ao verdadeiro do banco (ataque de *phishing*).
    - Como precaução o banco desligou durante uma manhã o seu *site*. (o ataque funcionou como fraude e DoS, com a "colaboração" do banco)

Técnicas de ataque

- Algumas vulnerabilidades não-tecnológicas
  - Doença de pessoal chave
  - Doença em simultaneo de muitos funcionários (por ex. Gripe)
  - Perda de pessoal chave (reforma, mudança de emprego, morte)
  - Perda de ligações telefónicas ou de rede
  - Perda de ligação de água, electricidade ou aquecimento/arrefecimento
  - Raio, inundação, fogo, tremor de terra, etc.
  - Falencia do vendedor ou produtor de computadores/software
  - Bugs de software
  - Greves

(extraido de “Practical UNIX & Internet Security”, por Garfinkel & Spafford)

Versão Web, FTP e servidor de *email*

- Os *hackers* tentarão descobrir quais as **versões das aplicações** referidas ligando-se às portas UDP e TCP e enviando dados aleatórios para cada uma.
- Alguns serviços respondem a dados aleatórios com um *banner* – dados que identificam a aplicação que está a correr e, eventualmente, a versão. Consultando bases de dados de vulnerabilidades poderão "afinar" o ataque. [http://www.securityfocus.com]

Informação sensível

- Se os *hackers* forem capazes de contactar certas portas (por exemplo TCP 139 ou 445) tentarão obter de forma anónima informação sensível do sistema incluindo:
  - Nomes de utilizadores
  - Últimas datas de *logon*
  - Datas de mudança de *passwords*
  - Filiação em grupos (*group membership*)
- Os *hackers* poderão utilizar os dados obtidos para realizarem ataques por força bruta para ganharem acesso aos sistemas. Por exemplo o *hacker* pode enumerar membros do grupo de administradores locais, procurando nomes como TEST ou BACKUP

Pontos a considerar

- Qual a informação que pode ser obtida nos portos à escuta? Que nível de permissões é necessário para enumerar esta informação?
- Existe um sistema de “logs” instalado para determinar se alguém tentou aceder a esta informação?
- Existe o potencial para um utilizador autenticado poder ver informação sensível ou informações pessoais que possam comprometer a privacidade?
- Qual a informação de “banner” que a aplicação fornece ao utilizador? Pode ser suprimida ou modificada pelo administrador do sistema?

Obter acesso via manipulação dos utilizadores

- Depois dos *hackers* obterem informações básicas suficientes sobre o seu alvo, tentarão ter acesso ao sistema alvo fazendo-se passar por utilizadores autênticos. Isto significa que necessitam da *password* para a conta de um utilizador que descobriram através dos passos anteriores. Existem duas formas de obter as *passwords*: usando engenharia social ou usando um ataque de força bruta.

Engenharia social

- É importante o que um empregado insuspeito fará por alguém que pareça autoritário. Alguns hackers utilizam a informação obtida a partir do registo de domínios ou do *site* Web da companhia e contactam directamente um empregado por telefone.
- Com alguma "conversa" conseguem convencer o empregado a revelar a sua *password* sem levantar suspeitas, fazendo-se passar, por exemplo, por alguém do *helpdesk*.

Ataques de força bruta

- Se a engenharia social não funcionar ou não for uma opção, existe sempre a aproximação pela força bruta. Estes ataques podem ser contra qualquer aplicação ou serviço que aceite autenticação dos utilizadores, incluindo (mas não limitadas a):
  - Network basic input/output system (NetBIOS) over TCP (TCP 139)
  - Direct Host (TCP 445)
  - Lightweight Directory Access Protocol (LDAP), (TCP 389)
  - FTP (TCP 21)
  - Telnet (TCP 23)
  - Simple Network Management Protocol (SNMP), (UDP 161)
  - Point-to-Point Tunneling Protocol (PPTP), (TCP 1723)
  - Terminal Services (TCP 3389)

Ataques de força bruta

- Se um *hacker* conseguir um dos serviços utilizará os nomes obtidos nos passos anteriores para lançar um ataque de força bruta. Para tal utiliza dicionários próprios para o efeito.
- Tipicamente os sistemas baseados em Windows não detectam este tipo de ataque dado a auditoria não estar activa por omissão. Excepto se as *passwords* forem muito complexas, estes ataques têm algum sucesso.
- Mesmo que os sistemas atacado façam *logs* destes ataques os atacantes tentam passar despercebidos e evitar a detecção utilizando nomes de servidores com caracteres ASCII não imprimiveis para aparecerem em branco nos *logs*.

Ferramentas mais usuais

Há muitos sites onde se podem obter ferramentas de teste de redes.
Alguns estão desatualizados outros nem por isso.

http://www.sectools.org

Pontos a considerar

- A auditoria dos *logins* falhados está activa por omissão?
- Existe algum mecanismo do lado do servidor que se possa usar para atrasar ou anular um ataque por força bruta?
- Consegue localizar a origem dum ataque de força bruta? Que informação consegue obter sobre a localização? Nome DNS ou endereço IP? Nome do computador? Endereço do *router* ou da máquina?
- Os atacantes conseguem alterar os *logs* do eventos ou de aplicações especificas depois de entrarem no sistema?
- Este protocolo necessita, por omissão, ser activado?

Escalar privilégios

- Após descobrirem uma *password* de uma conta de um utilizador e privilégios de acesso em modo utilizador os *hackers* irão tentar escalar as suas permissões. Usualmente começam por rever toda a informação sobre a máquina alvo a que têm acesso:
  - Ficheiros *batch* contendo nomes de utilizadores e *passwords* codificados são ouro para os *hackers*
  - Chaves de registos contendo *passwords* de utilizadores ou de aplicações também merecem uma espreitadela
  - A leitura de *email* ou outros documentos guardados no sistema pode também fornecer informação adicional aos *hackers* que lhes permita ganhar privilégios para noutros sistemas da rede

Escalar privilégios

- Se os *hackers* forem incapazes de enumerar informação estática útil do sistema podem prosseguir para a instalação dum "troiano". Isto envolve normalmente a instalação de código malicioso no sistema do utilizador e atribuir-lhe um nome duma aplicação utilizada frequentemente. Por exemplo, um *hacker* pode substituir o Notepad.exe por uma peça do código do "troiano" que torne alguém chamado Eric administrador do sistema antes de chamar o Notepad. A próxima vez que o utilizador se ligar ao sistema como administrador e chamar o Notpad é adicionada a conta para o Eric no grupo dos administradores de maneira transparente para quem chamou o Notepad.
- Se os *hackers* não puderem esperar que o utilizador utilize o Notepad, por exemplo, pode tentar escalar os privilégios através das aplicações do sistema que sejam executados com privilégios de administrador e substituir esta aplicação por um "troiano" que sirva para tornar Eric administrador. Quando o sistema arrancar de novo, o serviço irá correr e Eric passará a administrador.

Pontos a considerar

- Os utilizadores são capazes de aceder a informação sensível?
- As *passwords* das aplicações são guardadas de forma segura?
- As *passwords* são guardadas como texto em claro em ficheiros *batch*?
- Que chaves de registo (“registry keys”) podem os utilizadores comuns escrever? Algumas destas chaves são executadas com privilégios elevados?
- É possível a partir de contas de nível de utilizador modificar o contexto de segurança de serviços de tal forma que possam ser utilizados para lançar “troianos” com privilégios no sistema local?
- Existem alguns ficheiros que o utilizador possa sobrepor que sejam chamados por serviços a correr com privilégios elevados?

Obter *passwords* e segredos adicionais

- A primeira coisa que os *hackers* fazem após entrarem num sistema como administradores é obter o ficheiro de *passwords*. O ficheiro pode ser "tratado" com ferramentas como:
  - Pwdump2, para obter os *hashes* das *passwords,* e
  - LC3 ou John the Ripper, para "cracar" as *passwords*

[http://www.niatec.org , http://www.sectools.org]

- Obtendo *passwords* de sistemas locais, comparando-as e cruzando-as com utilizadores de outros sistemas o *hacker* pode chegar a administrador de domínio.
- Com tempo, paciência e sorte o *hacker, no pior caso,* pode conseguir entrar em todos os computadores do domínio.

Ferramentas mais usuais

Algumas das possíveis ferramentas para obter e “cracar” os ficheiros de *passwords* são:

- Pwdump2
- Lsadump2
- LC3
- John the Ripper

Pontos a considerar

- São gerados *logs* quando os ficheiros de *passwords* são acedidos?
- São gerados *logs* quando o administrador tenta um código errado para acesso aos dados das *passwords*?
- São guardadas no sistemas *passwords* para qualquer conta que possam ter maiores níveis de permissões que o administrador local?
- A *password* para as contas do nível de administração dum sistema são as mesmas que das contas de administrador noutros?
- Os utilizadores são encorajados a utilizar *passwords* complexas?

Instalação de *backdoors*

- Para o caso de terem de abandonar os sistemas comprometidos à pressa os *hackers* costumam criar *backdoors*. Podem tomar muitas formas mas a mais comum é deixar no sistema uma porta à “escuta” a partir da qual é possível aceder-lhe remotamente.
- *Firewalls* e filtragem nos *routers* pode evitar o acesso posterior aos sistemas. Uma filtragem menos eficiente pode deixar passar tráfego para portos como os TCP 20, 53, ou 8.

Backdoors - Reverse trafficking

- Forma complexa de *backdoor* que consiste num “troiano” que contacta o computador do *hacker* a intervalos de tempo bem definidos e regulares através do porto 80 do TCP. O computador comprometido pode disponibilizar um comando *shell* de nível sistema de maneira a permitir ao *hacker* executar código no computador comprometido.

- O *worm* Code Red usa esta técnica convencendo, através do porto 80 do TCP, servidores Web comprometidos a criarem ligações Trivial File Transfer Protocol (TFTP) (porto 69) do servidor para um máquina na Internet onde obtém código que interessa ao *hacker*.

Backdoors - Reverse trafficking

- Um *firewall* bem configurado pode não deixar criar ligações TFTP a partir de servidores Web ou de email para computadores que não sejam de confiança na Internet.

Ferramentas mais usuais

- Netcat

Pontos a considerar

- O sistema ou aplicação têm algum mecanismo para identificar código troiano que possa estar a correr no sistema?
- O sistema pode detectar dispositivos ou serviços que o atacante possa ter criado?
- Existe o conhecimento de portas conhecidas à escuta, serviços e dispositivos de maneira a\que o sistema possa ser monitorizado para ajudar a determinar se existe código pirata que tenha sido executado?
- Os dispositivos de segurança (*firewalls, routers*) estão configurados para prevenir tráfego de saída indesejado a partir de cada máquina?

Amplificar um sistema comprometido

- *Port redirectors*
  - Podem ser criadas num primeiro sistema comprometido e servirem para atacar outros sistemas, redireccionando as comunicações, escondendo a identidade do *hacker*, ultrapassando filtros em *routers* e *firewalls* e podendo até utilizar cifra na comunicação.

Amplificar um sistema comprometido

- Comprometer outros sistemas
  - Após comprometer um sistema e ter instalado os seus *backdoors* e seus *port redirectors* os *hackers* tentam comprometer outros sistemas na rede. Como o atacante pode agora trabalhar a partir dum sistema da interno de rede torna-se mais difícil detectar os ataques. Se não for apanhado antes de ganhar privilégios de administrador torna-se difícil expulsá-lo depois.

Ferramentas mais usuais

- Fpipe [http://www.mcafee.com/us/downloads/free-tools/index.aspx]

Pontos a considerar

- Existem processos para monitorizar os *logs* de múltiplos sistemas da rede e correlacionar sequências de ataque para sugerir que um ataque está em curso?
- São revistos a participação em grupos (“group membership”) numa base regular de maneira a assegurar que não foram adicionadas contas de *hackers* a grupos administrativos?