Uma Análise do Software de Segurança SATAN - Security Administrator Tool for Analyzing Netwoks

Leandro Márcio Bertholdo *Liane M. R. Tarouco*

Instituto de Informática
Universidade Federal do Rio Grande do Sul
Av. Bento Gonçalves, 9500 - Campus do Vale
91509-900 Porto Alegre, RS
E-mail:
berthold@inf.ufrgs.br liane@penta.ufrgs.br

Resumo

Em 5 de Abril de 1995 foi colocado a disposição da comunidade Internet um novo pacote para análise de segurança de sistemas e redes TCP/IP, o *software* SATAN. Desde antes da sua publicação esta ferramenta criou grandes controvérsias, sendo inclusive seus autores acusados de estar fornecendo para a comunidade hacker mais uma ferramenta de ataque. Agora, autorizada e em larga escala. O artigo aqui apresentado faz uma análise do software SATAN, quanto a sua forma de funcionamento, fraquezas, tipos de provas aos quais a ferramenta submete o sistema e mostra alguns pontos que a ferramenta não analisa e que são igualmente perigosos. Também estão descritos alguns dos problemas que foram detectados em testes realizados com sistemas reais.

Abstract

On 5 April 1995 a new package for systems and TCP/IP networks security analyzing, SATAN, was made available to all Internet Comunity. Before it has been publishing, this tool has already caused a great controversy: its authors were accused of exposing the net

Email, servidores de nomes para este domínio, e registros HINFO.

Um teste com o comando showmount também é realizado: "`showmount -e $target`" para verificar quais hosts montam que partição do host alvo. Outra tentativa é "`showmount -a $target`" para obter mais informações sobre o daemon mountd do host alvo, e aprender a respeito de outros hosts que possuam algum relacionamento (montam diretório ou tem diretório montado) sobre o alvo em questão, e que podem vir a ser considerados como novos alvos.

No nível *normal* são coletados banners sobre serviços bastante conhecidos, como FTP, Telnet, SMTP e outros. Estes dados são obtidos via duas ferramentas utilizadas pelo sistema: um scanner para portas TCP e um para portas UDP, cujo funcionamento será visto a seguir. Neste nível são identificados o tipo de sistema (SUN, SGI, DEC, IBM, HP), versões e existência dos serviços de rede que são mais comumente oferecidos (remote login, anonymous FTP, WWW, Gopher, Mail).

No mais alto nível (*heavy*), SATAN interroga o host à procura de falhas principalmente na parte de controle de acesso. Neste nível SATAN começa uma pesquisa a procura de usuários via comando "`rusers -l $target`", isto se o daemon ruserd estiver ativo (esta informação é obtida através da analise da saída do comando rpcinfo).

Outra observação realizada sobre a saída do comando rpcinfo é a respeito do serviço bootparam, responsável por prover informações para o boot de estações diskless. Através deste serviço podem ser obtidas informações junto ao servidor NIS, tais como arquivos de configuração do servidor, bem como o seu arquivo de passwords[HES].

Neste nível também é testado o comando finger, para obter maiores informações sobre determinado usuário. Este é um dos momentos onde SATAN poderia ter seu código mais facilmente modificado para implementar um quarto nível: tentar invadir o sistema. Aqui também é feito uma pesquisa em todos os números possíveis de portas, tanto TCP como UDP.

Os problemas mais comuns encontrados pelo software, segundo testes realizados pelos autores[VEN95], utilizando o nível máximo de pesquisa, são:

Sistema de arquivos (NFS) exportado para hosts específicos

```
Sistema de arquivos (NFS) exportado  para programas não privilegiados
Sistema de arquivos (NFS) exportado via portmapper
Arquivo de passwords do NIS acessível de hosts específicos
Versões antigas de Sendmail (anteriores a 8.6.10)
Acesso ao servidor REXD
servidor X sem controle de acesso
Acesso a determinados arquivos via TFTP
Shell remoto a partir de outros hosts
Diretório home do FTP anônimo aberto para escrita
```

3 A Estrutura de Funcionamento

SATAN é composto de um kernel central, que implementa a parte mais geral do sistema, e por vários outros pequenos programas responsáveis por detalhes de implementação de serviços de rede e testes de vulnerabilidades. O kernel é composto pelos seguintes subsistemas:

- **Cripto-ignição**: responsável pela geração de uma chave pseudo-randômica para validar a comunicação entre o cliente HTML e o servidor HTTP (Este servidor é implementado pelo próprio SATAN, e possui apenas um subconjunto dos comandos definidos pelo protocolo).
- **Máquina de política**: Este subsistema determina quais hosts serão pesquisados, e quais os níveis de teste são adequados para aquele hosts. Isto é realizado baseando-se no arquivo de configuração *satan.cf*. Aqui

Password System), do mesmo autor, onde os relatórios de vulnerabilidades do sistema nem sempre são claros e às vezes são até mesmo ignoradas devido à sua não compreensão pelos administradores de sistemas.

2.1 Níveis de Ataque

Como dito anteriormente, SATAN permite uma escolha entre diversos níveis de pesquisa tanto para hosts como para subredes. Dependendo da escolha, um conjunto de testes diferenciados é executado sobre os alvos escolhidos, abaixo estão especificados alguns testes demonstrados. Os níveis são cumulativos, ou seja, o nível máximo realiza todos os seus testes, além dos citados em normal e mínimo:

Nível	Aplicações Testadas
mínimo(0)	dns, rcp, showmount
normal(1)	tcpscan(portas conhecidas como WWW, gopher, ftp, ...), udpscan(53, 177), finger, rusers, boot, Yellow pages(NIS)
máximo(2)	tcpscan(1-9999), udpscan(1-2050, 32767-33500), NFS, rsh, ypbind, xhost, ftp

Figura 1: Testes realizados em cada nível de ataque

No nível mínimo(0), SATAN pesquisa o host alvo e estabelece as características gerais do sistema, se é um servidor de arquivos ou uma estação diskless. Neste momento são coletadas informações sobre o RPC (Remote Procedure Call). O sistema de RPC's é sondado via comando "`rpcinfo -p $target`" para criar uma lista de serviços RPC disponíveis. Esta lista de aplicações é armazenada em uma base de dados para ser utilizada posteriormente. Nesta primeira fase o DNS também é usado, através dele são verificados o nome e endereço ip dos alvos escolhidos. O seguinte comando é executado via script PERL**: "**`nslookup / set qt=any / $target`". Com ele são obtidas informações como hosts para troca de Email, servidores de nomes para este domínio, e registros HINFO.

Um teste com o comando showmount também é realizado**: "**`showmount -e $target`" para verificar quais hosts montam que partição do host alvo. Outra tentativa é "`showmount -a $target`", para obter mais informações sobre o daemon mountd do host alvo, e aprender a respeito de outros hosts que possuam algum relacionamento (montam diretório ou tem diretório montado) sobre o alvo em questão, e que podem vir a ser considerados como novos alvos.

No nível *normal* são coletados banners sobre serviços bastante conhecidos, como FTP, Telnet, SMTP e outros. Estes dados são obtidos via duas ferramentas utilizadas pelo sistema: um scanner para portas TCP e um para portas UDP, cujo funcionamento será visto a seguir. Neste nível são identificados o tipo de sistema (SUN, SGI, DEC, IBM, HP), versões e existência dos serviços de rede que são mais comumente oferecidos (remote login, anonymous FTP, WWW, Gopher, Mail).

No mais alto nível (*heavy*), SATAN interroga o host a procura de falhas principalmente na parte de controle de acesso. Neste nível SATAN começa uma pesquisa a procura de usuários via comando "`rusers -l $target`", isto se o daemon ruserd estiver ativo (esta informação é obtida através da analise da saída do comando rpcinfo).

Outra observação realizada sobre a saída do comando rpcinfo é a respeito do serviço bootparam, responsável por prover informações para o boot de estações diskless. Através deste serviço podem ser obtidas informações junto ao servidor NIS, tais como arquivos de configuração do servidor, bem como o seu arquivo de passwords[HES].

Neste nível também é testado o comando finger, para obter maiores informações sobre determinado usuário. Este é um dos momentos onde SATAN poderia ter seu código mais facilmente modificado para implementar um quarto nível: tentar invadir o sistema. Aqui também é feito uma pesquisa em todos os números possíveis de portas, tanto TCP como UDP.

Os problemas mais comuns encontrados pelo software, segundo testes realizados pelos autores[VEN95], utilizando o nível máximo de pesquisa, são:

```
Sistema de arquivos (NFS) exportado para hosts específicos
```

```
Sistema de arquivos (NFS) exportado  para programas não privilegiados
Sistema de arquivos (NFS) exportado via portmapper
Arquivo de passwords do NIS acessível de hosts específicos
Versões antigas de Sendmail (anteriores a 8.6.10)
Acesso ao servidor REXD
servidor X sem controle de acesso
Acesso a determinados arquivos via TFTP
Shell remoto a partir de outros hosts
Diretório home do FTP anônimo aberto para escrita
```

3 A Estrutura de Funcionamento

SATAN é composto de um kernel central, que implementa a parte mais geral do sistema, e por vários outros pequenos programas responsáveis por detalhes de implementação de serviços de rede e testes de vulnerabilidades. O kernel é composto pelos seguintes subsistemas:

- **Cripto-ignição**: responsável pela geração de uma chave pseudo-randômica para validar a comunicação entre o cliente HTML e o servidor HTTP (Este servidor é implementado pelo próprio SATAN, e possui apenas um subconjunto dos comandos definidos pelo protocolo).
- **Máquina de política**: Este subsistema determina quais hosts serão pesquisados, e quais os níveis de teste são adequados para aquele hosts. Isto é realizado baseando-se no arquivo de configuração *satan.cf*. Aqui é definida a política para a análise de segurança a ser usada pelo SATAN.
- **Aquisição de alvos**: dada uma lista de hosts, este subsistema gera uma lista de testes a serem utilizados para aqueles hosts
- **Aquisição de dados**: A partir da lista de testes gerada pelo subsistema de aquisição de alvos, este subsistema executa o conjunto de operações correspondente, e gera novos fatos, que servirão como entrada para a máquina de inferência
- **Máquina de inferência**: responsável por gerar uma nova lista de hosts, testes e fatos, baseada nos dados obtidos pelo sistema
- **Relatórios e análise**: subsistema responsável pela tabulação dos dados na linguagem HTML, para ser repassados ao browser WWW. É composto basicamente por scripts PERL, que lêem os arquivos de resultado dos testes e os formatam em textos HTML.

SATAN opera iniciando conexões para diferentes portas em computadores remotos, e então determina que tipo de informação cada porta retorna. Ele testa um grande número de vulnerabilidades, incluindo NFS (Network File System) e NIS (Network Information Service).

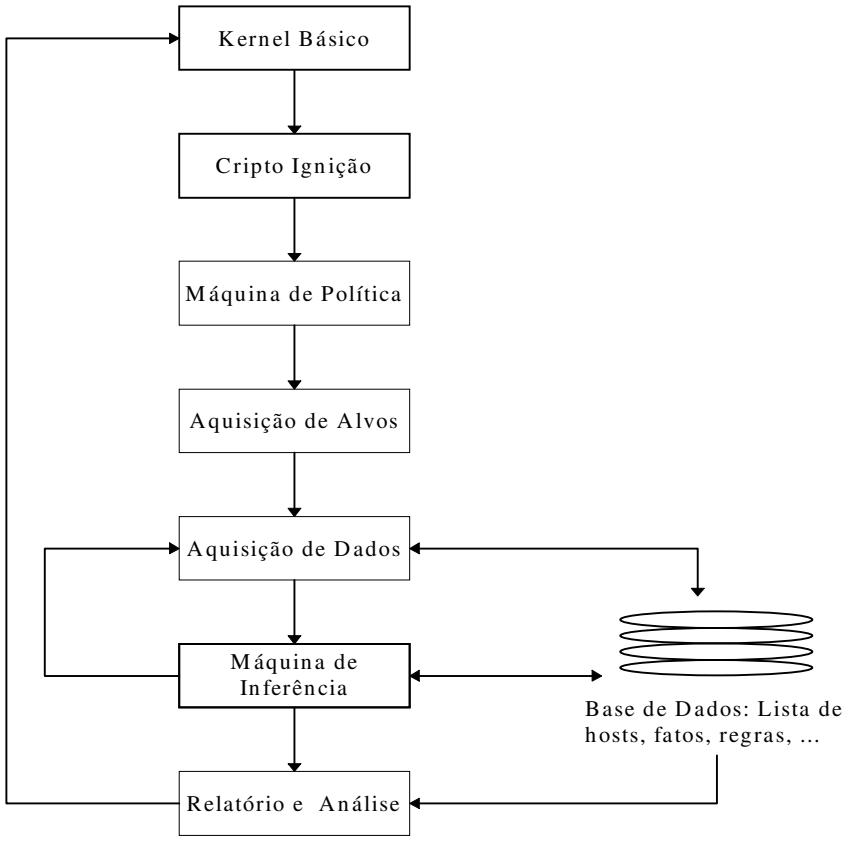


Figura 2: A estrutura do sistema

3.1 Os Arquivos de Regras

Tendo em vista que regularmente são descobertas novas brechas de segurança, o programa SATAN se utiliza de uma sistemática de regras para definir o que pode ser considerada a parte mais mutável em um sistema de segurança: as condições que especificam o que é seguro e o que é sabidamente vulnerável. Desta forma o sistema tem condições de se adaptar às novas falhas de segurança, sem necessitar de maiores alterações no código já construído. Usando esta metodologia, uma nova falha pode ser implementada simplesmente inserindo-se uma nova regra no sistema e criando-se um novo módulo (.satan) para avaliar as condições que definem a vulnerabilidade. Existem implementados cinco arquivos de regras, abrangendo diferentes pontos de configuração, listados abaixo (segue um exemplo de regras a cada arquivo):

- **Hosttype**: Regras que reconhecem banners para diferentes aplicações, como TELNET, FTP e SENDMAIL.

```
CLASS HP
/(HP-UX) .+ ([AB1-7]\.[A-Za-z0-9.]+) /                    "$1 $2"
UNKNOWN && /(HP-UX)/
UNKNOWN && /HP Sendmail/                                  "HP-UX"
```

- **Drop**: Determina quais fatos devem ser ignorados durante a avaliação do sistema, como um CDROM exportado para o mundo.

```
$text =~ /exports \/cdrom/i
```

- **Facts**: Regras que deduzem novos fatos a partir de dados existentes.

```
/runs rexd/     $target|assert|a|us|ANY@$target|ANY@ANY|REXD access|rexd is
vulnerable

/sendmail 8\.6\.([0-9]+)/i && $1 < 10 \
                $target|assert|a|rs|ANY@$target|ANY@$target|Sendmail
vulnerabilities|sendmail pre 8.6.10

/HP Sendmail \(1\.37\.109\.11/ \
                $target|assert|a|rs|ANY@$target|ANY@$target|Sendmail
vulnerabilities|sendmail pre 8.6.10
```

- **Services**: Regras que classificam os hosts por serviços por eles providos. Estas regras são reponsáveis pela tradução dos registros de dados internos do SATAN em informações a serem mostradas nos relatórios para os usuários.

```
/\/root\/\S+ to \S+/                    NFS (diskless)
/offers gopher/                         Gopher
/offers http/                           WWW
/offers X-[0-9]+$/                      X Windows
$service eq "xdmcp"                     XDM (X login)
```

- **Todo**: Regras que especificam quais testes devem ser realizados, e a ordem das ferramentas a serem disparadas. Como a definição das regras é realizada como sendo a definição de um script PERL, os dados obtidos por uma ferramenta podem servir como entrada de dados para outra, via uso de variáveis do script.

```
$service eq "mountd"                    $target "showmount.satan"
$service eq "ypserv"                    $target "ypbind.satan"
$service eq "rexd"                      $target "rex.satan"
$service eq "rusersd"                   $target "rusers.satan"
$severity eq "l"&&"$trustee|$trusted" =~ /(.*)@.*@(.*)/\$2 "finger.satan""-u
$1"
$service eq "ftp"                       $target "ftp.satan"
$untrusted_host && $service eq "shell"  $target "rsh.satan"
$untrusted_host && $service eq "shell"  $target "rsh.satan" "-u root"
$service eq "tftp"                      $target "tftp.satan"
$untrusted_host && /IRIX/               $target "rsh.satan" "-u guest"
```

- **Trust**: Regras que classificam os Hosts através da sua relação de confiança. Estas regras também são usadas para tradução entre as formas de armazenamento internas do SATAN para informações classificadas, compreensíveis pelo usuário.

```
$severity eq "l" && $trustee =~ /^root@/        root login
$severity eq "l" && $trustee !~ /^root@/        user login
$text =~ /exports \S+ to/                       NFS export
$text =~ / mounts \S+/                          NFS export
/serves nis domain/                             NIS client
$trustee =~ /\/export\/root\/(.*)@(.*)/         boot client
```

3.2 Os Scripts de Teste

O subsistema de aquisição de dados e a máquina de inferência baseiam-se em vários scripts processados pela aplicação SATAN para localizar e inferir sobre possíveis falhas na segurança. Estes scripts são construidos em linguagem PERL, e cada um é responsável por testar diferentes aplicações ou obter mais informações para o sistema de inferência, são eles:

- **boot.satan** : Através dele podem ser obtidas informações sobre o domínio de nomes do NIS. Usa o comando `bootparam` para obter informações de estações que permitem o boot de estações diskless.
- **dns.satan**: Usa o servidor de nomes para conversões nome->endereço e vice-versa (usa nslookup). Novos alvos também são localizados através deste serviço.

- **finger.satan**: É usado para obter informações junto ao daemon da máquina alvo sobre usuários logados, e apartir de que hosts. Os host aqui descobertos também podem ser usados para realimentar a lista de hosts do sistema.
- **ftp.satan**: Verifica se o serviço provê "anonymous ftp"(frequentemente mal configurado) , ou se o diretório home do usuário ftp está aberto para escrita (permite a execução remota de comandos, via arquivos como `.forward`).
- **nfs-chk.satan**: Consulta o portmapper e aponta sistemas de arquivos que podem ser montados por programas não privilegiados.
- **rex.satan**: Procura pela existência do daemon rexd (execução remota de comandos), e noticia como uma falha na segurança.
- **rpc.satan**: Faz uma consulta ao portmapper via "`rcpinfo -p`" para descobrir quais são as aplicações que usam RCP e que estão sendo usadas pela máquina alvo.
- **rsh.satan**: Verifica se o host destino confia no host a partir do qual SATAN está executando. Isto depende de arquivos de configuração como `.rhosts` e `hosts.equiv` [GAR91].
- **rusers.satan**: Tenta obter nomes de usuários que estão logados na máquina, podendo também obter mais nomes de hosts para a sua lista de alvos.
- **showmount.satan**: Tenta obter informações sobre systemas de arquivos exportados para todo o mundo sem restrição.
- **tcpscan.satan**: Reporta os serviços TCP usados. Invoca a ferramenta TCP_SCAN.
- **tftp.satan**: Conecta o hosts tentando pegar o arquivo `/etc/passwd`. Se o serviço estiver mal configurado, o arquivo será obtido
- **udpscan.satan**: Reporta os serviços UDP que estão sendo usados. Usa a ferramenta UDP_SCAN.
- **xhost.satan**: Verifica o display das aplicação X, e analisa se ela está vulnerável a alguns tipos de ataque mais simples.
- **yp-chk.satan**: Pesquisa o servidor NIS. Tenta encontrar por exemplo arquivos de configuração, como o de passwords, acessíveis.
- **ypbind.satan**: Tenta descobrir qual é o domínio de nomes do NIS, bem como qual é o seu servidor.

Existem outros programas responsáveis por descobertas de informações sobre a rede que não são citados aqui, como o *fping*, mostrado a seguir.

4 A Descoberta dos Alvos

O primeiro passo para qualquer programa que faça uma análise de uma rede é descobrir quais são os componentes da rede. SATAN não poderia ser diferente; ele se utiliza de ferramentas como ***nslookup*** (para consultas ao DNS) e ***fping*** (uma variação do programa ping).

Todo o subsistema de aquisição de alvos esta baseado principalmente no programa fping, que é usado para descobrir todos os hosts ativos na subrede onde o alvo primário foi definido. ***fping*** é um variação realizada sobre o programa PING de maneira a aumentar a eficiência em relação ao ao seu antecessor, ele também se baseia no protocolo ICMP, enviando pedidos e aguardando suas respostas. A diferença entre as duas implementações é que ***fping*** aceita uma lista de nomes/endereços de hosts a serem verificados, na forma:

```
fping HOST1 HOST2 ... HOSTn
```

Ao invés de testar um hosts por vez enviando um request e esperando pela respectiva resposta ou um *timeout* do destino, ***fping*** envia um pacote 'ping' para HOST1, e vai para o

próximo da lista, seguindo um escalonamento do tipo 'round-robin'. Quando um host envia uma resposta, ela é anotada, e o nome do host apagado da lista de escalonamento. Se não houver uma resposta dentro de um limite de tempo previamente estabelecido e/ou dentro do limite estabelecido para uma retentativa, o host será considerado como '*unreachable*'.

SATAN passa para a aplicação ***fping*** uma lista de 256 hosts, independente da máscara usada pela rede. As tentativas são fixas, como se fosse uma rede classe 'C'. Além do aumento na eficiência da pesquisa, uma outra vantagem obtida via ***fping*** é a possibilidade de se verificar hosts através de pontes e roteadores, o que não é conseguido com técnicas de inundação (pacotes de broadcast).

4.1 O Subsistema de Aquisição de Dados

Após a obtenção e validação de uma lista de hosts, o subsistema de aquisição de alvos repassa o controle ao subsistema de aquisição de dados, responsável por obter informações sobre um determinado host. Este subsistema visa obter informações sobre quais aplicações estão sendo executadas na máquina alvo, isto é realizado analisando todas as portas de um determinado host. Elas são analisadas a partir de duas ferramentas: TCP_SCAN e UDP_SCAN. Estas ferramentas procuram por todos os serviços disponibilizados pelo host para a rede. SATAN permite especificar intervalos de portas a serem avaliadas (vide intensidade de ataque), visando com isso aumentar o desempenho do sistema. Todo o processo de verificação das portas é realizado em paralelo.

A verificação e validação das portas usadas pelo sistema é uma maneira de localizar em primeiro lugar quais os serviços são fornecidos por um determinado host, e assim identificar possíveis falhas no sistema, como encontrar um host que fornece serviços como TFTPD e REXD, cuja simples existência já é potencialmente uma falha na segurança do sistema. Outra brecha na segurança que pode ser localizada a partir deste método é a localização de programas não identificados instalados em portas desconhecidas, isto é, nenhuma daquelas listadas na última RFC [REY94] *Assigned Numbers***.** Possivelmente, *back doors* padrões usadas pela comunidade hacker, e devidamente documentadas na sua própria RFC - "Resource For Crackers" [BEL93].

4.2 Descobrindo Portas TCP

O programa TCP_SCAN é usado para determinar a disponibilidade dos serviços TCP em um dado host. Opcionalmente ele pode também ser usado para obter banners de aplicações. Inicialmente o programa aloca um socket para a leitura das respostas ICMP obtidas. A seguir são testados todos os números de portas passados como parâmetros. As portas a serem selecionadas estão diretamente relacionados com o nível de pesquisa configurado pelo usuário (0-mínimo,1-normal, 2-máximo).

Mímino: não usa
Normal: Portas do Gopher(70),http(80),ftp,telnet(23),SMTP(25),NNTP ,UUCP e X (6000)
Máximo: Todas as portas entre 1-9999, além das portas do nível normal

Uma vez estabelecida a conexão com o HOST:PORTA destino, o programa tenta obter durante um intervalo de tempo pré-estabelecido o banner da aplicação que está instalada naquele número de porta. Após a recepção deste banner ou a obtenção de um time-out da porta, o TCP_SCAN envia um 'QUIT\r\n' e encerra a conexão. Todas as informações obtidas são armazenadas na base de dados, para serem posteriormente apresentadas ao usuário, através do subsistema de análise e relato de dados.

4.3 Descobrindo Portas UDP

A pesquisa através das portas UDP é um pouco mais trabalhosa. Como o protocolo UDP não implementa nenhum tipo de controle de congestionamento ou retransmissão de pacotes, o programa precisa estabelecer um controle mínimo sobre estes itens, já que podem ser gerados aproximadamente 2800 pacotes UDP para as diferentes portas de cada host a ser analisado, e cujas respostas são dependentes das velocidades de processamento dos hosts e do segmento de rede entre eles.

Para este controle o programa se baseia em uma janela de transmissão que tem sua abertura diminuída quando forem constatadas as primeiras retransmissões realizadas.

O número de portas UDP a ser analisado também é dependente do nível ao qual o sistema está sendo provado:

Mínimo: não usa
Normal: Avalia as portas 53 e 177
Máximo: Todas as portas entre 1-2050, e entre 32767-33500, além das portas do nível normal

Após o envio de um pacote do tipo '0\r\n' para HOST:PORTA destino, o programa aguarda a resposta na tentativa de obter mais informações sobre aquela porta. No caso do UDP_SCAN podem haver várias retentativas até que se esgote o tempo máximo de espera pela resposta, a critério da subrotina de controle de fluxo. Todas as Informações obtidas são armazenadas na base de dados para serem posteriormente apresentadas ao usuário, através do subsistema de análise e relato de dados.

4.4 Teias de Confiança

O software SATAN também permite obter relatórios quanto à teias de confiança [CHE94]. Ele permite gerar informações a respeito de uma subrede, relatando quais são os hosts considerados pela maioria como confiáveis. Um host considerado confiável possui privilégios como: login sem a exigência de senhas (*root* ou usuários normais) e compartilhar dados (diretórios montados remotamentes - NFS) das máquinas que confiam nele.

SATAN obtém informações sobre as relações de confiança do sistema baseado principalmente nas informações obtidas sobre o sistema de RPC's. Em hipótese alguma ele tenta invadir uma máquina para obter arquivos como o `hosts.equiv` ou `.rhost`, mesmo que muitas vezes isto significasse apenas mais uma linha de código. Informações sobre relações de confiança são frequentemente usadas por hackers para definir quais alvos são os mais importantes da rede; a partir dos quais a maior parte da rede ficará comprometida.

5 Utilizando o SATAN

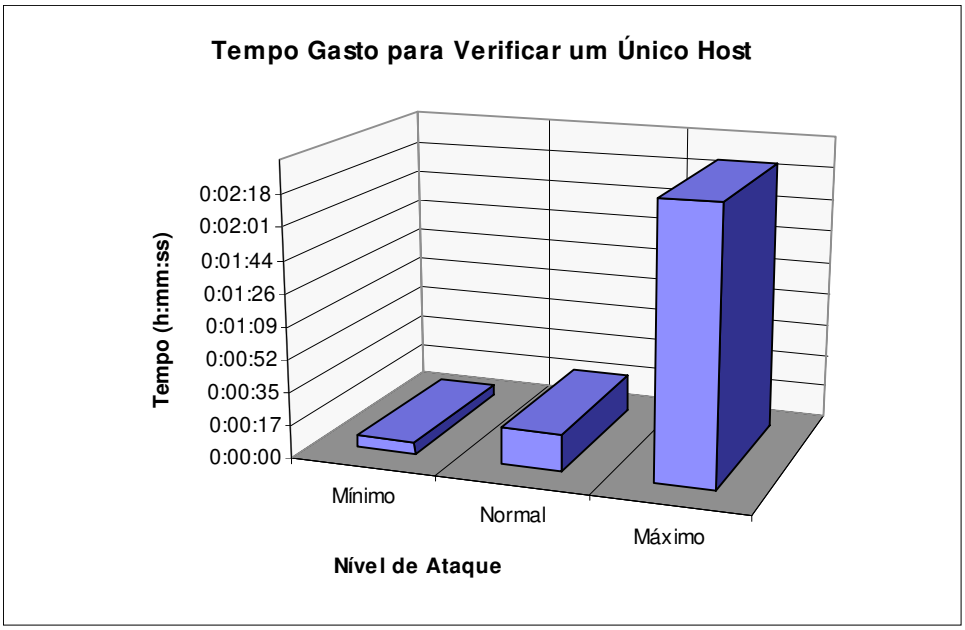
O software SATAN foi utilizado em uma rede, que vai ser referida aqui como XPTO, para avaliar o nível da segurança de todo o domínio xpto.br, maiores informações podem ser obtidas no relatório em anexo.

O software conseguiu obter sucesso em várias tentativas de burlar a segurança da rede inspecionada. Dentre os pontos testados, o conjunto de fraquezas encontradas foram:

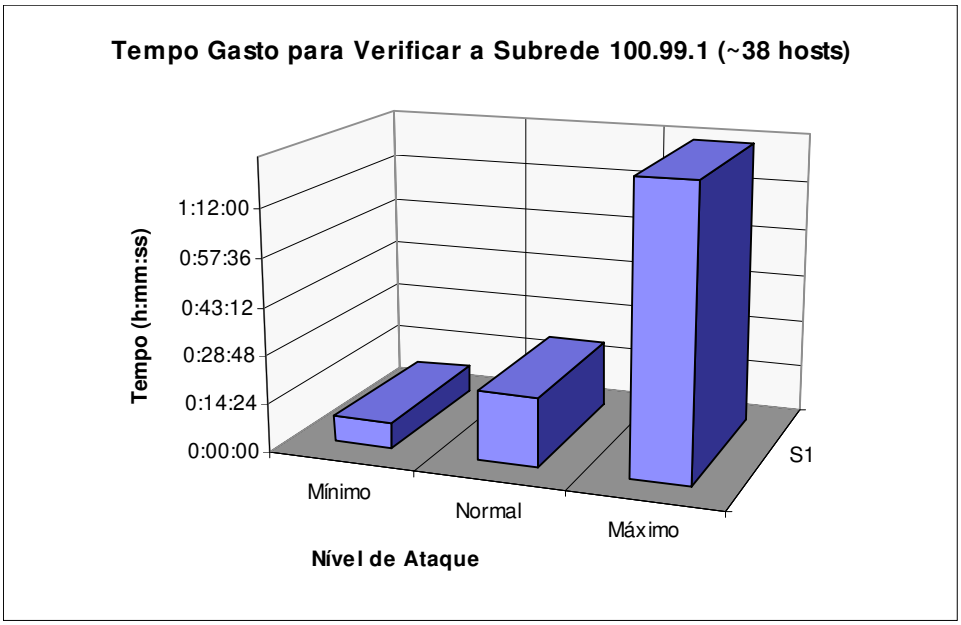
```
Sistema de arquivos (NFS) exportado para hosts específicos
Sistema de arquivos (NFS) exportado  para programas não privilegiados
Sistema de arquivos (NFS) exportado via portmapper
servidor X sem controle de acesso
Acesso a determinados arquivos via TFTP
Shell remoto a partir de outros hosts
```

Alguns destes permitiam ter total controle sobre segmentos da rede. Abaixo estão alguns resultados estatísticos obtidos pelo uso do software, e que esperamos, ajudem futuros administradores a conseguir os melhores resultados e no menor tempo possível.

O gráfico abaixo mostra o tempo que foi gasto para pesquisar as vulnerabilidades da máquina alvo. Como pode se observar, utilizar o nível máximo (heavy) de scanning, tem o seu preço, e muitas vezes uma eficiência igual.

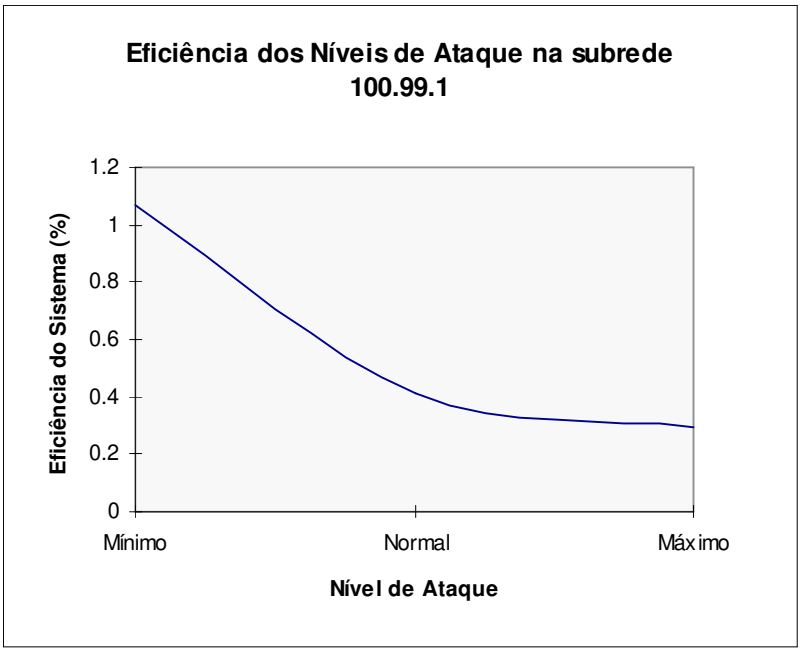


O gráfico abaixo mostra uma relação entre tempo e nível de ataque selecionado, só que agora considerando uma subrede (aqui representada por um número fictício 100.99.1).



Todas as informações obtidas serviram para nos levar a um último gráfico, o da eficiência do sistema, que nos deveria dar uma ajuda a ter uma resposta para a seguinte pergunta:

O nível mínimo pode ser satisfatório se for necessário fazer-se uma pesquisa em toda a rede da instituição da forma mais rápida possível? Já que uma pesquisa no nível máximo poderia levar dias.



Na amostragem realizada na subrede 1 utilizando o nível mínimo, foram encontradas 5 vulnerabilidades, no nível normal nenhuma a mais foi encontrada, enquanto o nível máximo havia encontrado cerca de 15 falhas na segurança.

Através do Software SATAN foram apontadas mais de 70 vulnerabilidades entre os aproximadamente 150 hosts pesquisados. Logicamente nem tudo o que foi apontado foi considerado como vulnerabilidade, ou considerado realmente como um problema de segurança. Havia entretanto, casos onde um invasor poderia facilmente obter o controle como *root* em partes importantes da rede da instituição.

6. Conclusão

SATAN é uma poderosa ferramenta para avaliar a segurança de um *site*, principalmente para uma universidade, onde por questões filosóficas isolar a rede com firewalls geralmente é inviável; assim como utilizar roteadores como filtros de pacotes. Neste caso, o programa SATAN é uma maneira de aumentar o nível médio de segurança de todo o *site*.

A segurança é na prática dependente dos padrões e conhecimentos de segurança dos inúmeros administradores existentes no *site*. Em alguns ambientes heterogêneos em termos de usuários e máquinas, a segurança pode se degradar com o tempo. Em outros ambientes, que são palco de experimentos; pessoas com diferentes conhecimentos e interesses, instalam diariamente inúmeras aplicações, máquinas e sistemas operacionais inovadores. Estes mesmos usuários dificilmente se preocupam com problemas relativos à segurança, repassando esta tarefa ao administrador responsável pela rede, que nem sempre está ciente da complexidade de tal tarefa.

Neste cenário, uma ferramenta como o software S.A.T.A.N, representa um auxílio importante para que os administradores possam realizar periodicamente uma verificação de segurança padrão sobre problemas bem conhecidos por hackers, mas geralmente ignorados por administradores menos experientes.

Bibliografia

[BEL93] BELLOVIN, Steven M.. **Packets Found on an Internet.** Computer Communications Review, julho de 1993, Vol. 23, No. 3, pg. 26-31.

[BRY94] BRYANT, Richard Bringle**. Unix Security for the Organization: Implement a Secure Environment - and Keep It That Way.** SAMS Publishing, 1994.

[CHE94] CHESWICK, William R.; BELLOVIN, Steven M..**Firewalls and Internet Security: Repelling the Wily Hacker.** Addison-Wesley Publishing Company. 1994.

[FAR95] FARMER, Dan; VENEMA, Wietse**. Improving the Security of Your Site by Breaking Into it.** Intenet Security, march 1995, pg. 4-13.

[GAR91] GARFINKEL, Simson; SPAFFORD, Gene. **Practical Unix Security.** O'Reilly & Associates, Inc. 1991. Sétima edição 1994.

[HES] HESS, David K.; SAFFORD, David R.; POOCH, Udo W..**A Unix Network Protocol Security Study: Network Information Service.**Texas A&M University.

[STA95] STALLINGS, William. **Network and Internetwork Security - Principles and Practice.** Prentice Hall International Editions.New Jersey, 1995.

[VEN95] VENEMA, Wietse. S.A.T.A.N. **A Summary. Internet Security.** march 1995, pg. 3

[RP90] REYNOLDS, Joyce K.; POSTEL, Jon B. **Request For Comments 1060: Assigned Numbers**, March 1990.