

Clube dos Mercenários

<http://cdm.frontthescene.com.br/>

OBS: Esboço inicialmente escrito para a Conferência H2HC, não ocorreu(15/07/2004), veja .ppt que acompanha este arquivo.

Breaking NIDS

Por Glaudson Ocampos

nashleon@yahoo.com.br

Base de Pesquisa:

(<http://ouah.kernsh.org/network-intrusion-detection.html#9.3>)

(<http://www.robertgraham.com/pubs/network-intrusion-detection.html>)

(<http://www.phrack.org/>)

(<http://www.google.com.br/>)

Tópicos

Introdução a NIDS

Detecção de Regras

Técnicas de Evasão

Exemplos de Ataques

Outros Problemas em NIDS

Ferramentas Públicas

O Futuro das NIDS

Conclusão

Créditos

Introdução a NIDS

Sistemas de Detecção de Intrusos (IDS) é uma tecnologia de segurança que procura identificar e isolar tentativas de invasão em sistemas de computador.

De um modo geral, toda tecnologia que visa detectar ataques que ocorreram e/ou tentativas de ataques recebe hoje o nome de IDS.

Existem várias classificações de IDS e define-se como NIDS, os Sistemas de Detecção de Intrusos que atuam a nível de rede. Através de uma interface executando em modo promíscuo, um NIDS é capaz de monitorar uma rede em tempo real no intuito de detectar tentativas de invasão, ataques oriundos de worms e também anomalias na transmissão de dados da rede.

Alguns Softwares NIDS muito usados são o Snort, Real Secure, Prelude e etc.

A grande maioria dos NIDS utilizam alguns conceitos para a detecção de ataques. Estes conceitos são diversos, mas podemos enumerar alguns:

- Assinatura de Ataques -> Com base nos exploits criados e métodos padrões de ataque, um NIDS utiliza um sistema de regras(assinaturas) capaz de detectar quando um ataque está ocorrendo;
- Anomalia na Rede -> Com base em padrões de protocolo e serviços, um NIDS é capaz de detectar anomalias no serviço através de dados estranhos transitando pelo mesmo (por exemplo, ICMP túnel);
- Falhas de Autenticação -> Sucessivas falhas na autenticação de um usuário(string de erro de autenticação num curto período de tempo), podem alarmar um NIDS, fazendo com que o mesmo detecte um ataque de Força Bruta.

Detecção de Regras

Os ataques do tipo “Detection” compreende a capacidade de detectar quais regras, configurações e uso um determinado sistema de detecção de intrusos possui em uma determinada rede alvo.

Existem sistemas de IDS que “derrubam” a conexão (Active Sniffing/IDS) procurando impedir ataques em “Run-time”. Assim o atacante passa a saber que no “alvo” existe um sistema de monitoração.

Esta característica torna o sistema propenso a detecção de regras com exatidão por

parte do atacante, onde enviaria um determinado ataque e se a conexão permanecesse ativa, ele então saberia se determinada regra está ativa e também quais meios empregar para ser bem sucedido.

Um exemplo de Active/IDS barrando um ataque pode ser visto abaixo:

```
root@kimera:~# telnet localhost 21
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 ProFTPD 1.2.5 Server (ProFTPD Default Installation)
[kimera.localdomain]
USER AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Connection closed by foreign host.
```

Acima vemos uma tentativa de exploração de um servidor ProFTPd onde um atacante tenta enviar NOPS ou PADDINGS através do comando USER e sua conexão é devidamente barrada.

Técnicas de Evasão:

+ Fragmentação -> É a habilidade de dividir um único pacote IP em múltiplos pacotes menores(quando usado com “low”[transmissão lenta]) se torna mais letal ainda). Muitos softwares e sistemas operacionais tem problemas ao manusear fragmentos de pacotes, e ataques recentes podem ser visto no FreeBSD e LibNIDS.

+ De-sincronização -> É um ataque antigo que explora fraquezas nos modelos TCP/IP dos NIDS.

Existem vários métodos para executar este tipo de ataque, um deles consiste no atacante inserir caracteres extras sobre uma transação, que serão inválidos e não serão processados pelo host, mas que os NIDS poderão aceitar os caracteres e falhar em ver o que realmente está acontecendo.

Um exemplo básico consiste em enviar a seguinte sequência de pacotes:

GET /cgi-b – (Pacote Válido)

NASH - (Pacote Inválido)

in – (Pacote Válido)

O NIDS interpreta como sendo “GET /cgi-NASHin” já o sistema operacional

interpreta "GET /cgi-bin".

Existem várias formas para enviar um pacote inválido ao host destino e que o NIDS pode interpretar, alguns deles são:

- Inserir Dados com Número de Sequência Errado;
- Spoofar pacotes FIN/RST com Número de Sequência Errado;
- Inserir Dados com TCP Checksum Errado;
- Spoofar pacotes FIN/RST com TCP Checksum Errado;
- Inserir Dados com TTL curto;
- Spoofar pacotes FIN/RST com TTL curto;

Alguns ataques de de-sincronização são complexos e necessitam de um suporte no kernel para funcionar a contento.

+ Avoiding Defaults -> Um atacante pode instalar uma backdoor conhecida, por exemplo NetBUS, em uma porta desconhecida, inutilizando a regra padrão para NetBUS que seria a porta 12345 ou 12346.

```
alert tcp $HOME_NET 12345:12346 -> $EXTERNAL_NET any (msg:"BACKDOOR
netbus active"; flow:from_server,established; content:"NetBus";
reference:arachnids,401; classtype:misc-activity; sid:109; rev:4;)
```

+ Slow Scans -> Por causa do volume de tráfego na rede, um NIDS precisa não gerar dados muito lentos no log (Um portscan muito demorado, poderá passar despercebido por uma ferramenta NIDS).

Tabela de Um Ataque Comum Sendo Detectado

Horário	Comando	Log no NIDS	Tempo de Permanência no Buffer a Espera de Incrementação
11:32	nmap -sF -p 21 <host_alvo>	10.24.0.5 -> 21	5 minutos*
11:32	nmap -sF -p 22 <host_alvo>	10.24.0.5 -> 21 10.24.0.5 -> 22	5 minutos*
11:33	nmap -sF -p 23 <host_alvo>	10.24.0.5 -> 21 10.24.0.5 -> 22 10.24.0.5 -> 23	5 minutos*

Tabela de Um Ataque Slow Scan Passando Pela Detecção

Horário	Comando	Log no NIDS	Tempo de Permanência no Buffer a Espera de Incrementação
14:08	nmap -sF -p 21 <host_alvo>	10.24.0.5 -> 21	5 minutos*
17:43	nmap -sF -p 22 <host_alvo>	10.24.0.5 -> 22	5 minutos*
20:15	nmap -sF -p 23 <host_alvo>	10.24.0.5 -> 23	5 minutos*

Uma Ferramenta NIDS que trabalha em cima deste conceito é o PortSentry.

+ Ataques Coordenados -> Scan de múltiplos IPs dominados pelo atacante, não gerando alarmes no sistema (fazendo poucas requisições por máquina (IP) dominado) e redirecionando qualquer forense para máquinas de terceiros.

+ Spoof -> Smurf e ISN podem ludibriar o sistema de IDS (LD-Scanning), gerar muitos logs falsos (Decoy Scan) dificultando qualquer posterior Análise de Log e Forense.

O NIDS pode ainda não detectar ataques pois imagina que estão vindo de uma interface externa:

```
alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:"DOS Teardrop attack";
id:242; fragbits:M; reference:cve,CAN-1999-0015;
reference:url,www.cert.org/advisories/CA-1997-28.html;
reference:bugtraq,124; classtype:attempted-dos; sid:270; rev:2;)
```

* teardrop = ataque utilizando fragmentação de pacotes, fazendo reassembly ocorrer de maneira inesperada;

\$EXTERNAL_NET pode ser definido como qualquer computador diferente de \$HOME_NET, no entanto o atacante pode enviar pacotes fazendo com que os endereços de origem apontem para um IP da HOME_NET, executando assim um ataque, por exemplo, do tipo teardrop ludibriando o sistema.

Se o atacante estiver no mesmo segmento de rede ou na rede interna, ele pode utilizar os conceitos de LD_SCANNING para gerar logs falsos, e pode até mesmo utilizar IP Hijacking (ARP Poison) para entupir o sistema NIDS com dados falsos.

+ Proxy -> Um atacante pode camuflar seu endereço real, dificultando ao sistema de

segurança saber quem(de onde) está atacando realmente. A quantidade de proxies ativos e abertos na Internet propicia ao atacante fazer conexões legítimas camuflando seu real IP. Outro ponto importante reside no uso de Robots para a execução de determinados ataques, camuflando o endereço para o endereço de um Robot e em muitos casos gerando muitas informações falsas no sistema de log.

+ Pattern Match Evasion -> Quebra do sistema de assinatura(regras) de exploits, URL, etc, com base nas possibilidades de representação diferenciada de dados.

Exemplos de Ataques

(Regras no Snort)

Alguns dos ataques mais comuns utilizados em sistemas Internet (TCP/IP) podem utilizar os conceitos de evasão para passarem despercebidos pelos NIDS e não serem barrados. Neste item veremos alguns exemplos de ataques comuns e alguns problemas que os sistemas mais comuns de NIDS enfrentam. Os exemplos serão para o Snort, mas são expansíveis a qualquer sistema NIDS que utiliza conceitos semelhantes de detecção de ataques.

Ataques de Buffer Overflows

OBS: Shellcodes são pequenas instruções assembly que são usados para executar uma shell ou comandos shell, tipicamente como um resultado de um ataque de buffer overflow.

Buffer Overflow é uma técnica que é usada para descrever o ato de encher um pedaço de memória com mais espaço do que ela suporta, permitindo muitas vezes a um atacante alterar o fluxo de execução de um programa.

- Regras Contra NOPs (inclui NIDSFindShellcode);

É muito comum um NIDS possuir regras para detecção de instruções vazias(NOPS) em uma base de dados. Mas o que são NOPS ou instruções vazias?

Em determinados casos, um atacante pode tentar obter acesso a um sistema através de falhas conhecidas como Buffer Overflows e Format Strings. Ataques de Buffer Overflows exploram falhas no designer de um programa que permite a um atacante obter o controle da execução do mesmo. Quando um atacante obtém o controle da execução de um programa, por exemplo, de um servidor, ele procura na maioria das vezes obter acesso a uma shell remota que proporcionará maiores condições para que ele venha dominar toda a máquina ou todo o sistema operacional, mais precisamente elevar o privilégio para super-usuário.

No entanto, em alguns casos a exploração via Buffer Overflows ou Format Strings requer a utilização de instruções vazias (NOPs), encontradas no buffer a ser usado no redirecionamento da execução. Um atacante pode utilizar uma cadeia de NOPs para ser bem sucedido neste tipo de exploração.

Sabendo disso, alguns administradores de NIDS podem resolver criar regras para detectar a passagem de instruções vazias pela interface de rede, conseguindo assim detectar tentativas de ataques de Buffer Overflows, sem levar em conta um exploit público conhecido, mas visando a técnica de escrita de exploits para buffer overflows mais conhecida.

A instrução vazia(NOP) e seu representante em hexa na arquitetura Intel x86 é 0x90(90H).

Logo, uma base de dados de assinaturas de um NIDS pode conter regras para detectar tal instrução vazia. Abaixo vemos o Snort contendo tais regras:

```
alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any (msg:"SHELLCODE
x86 NOOP"; content: "|90 90 90 90 90 90 90 90 90 90 90 90 90 90|"; depth:
128; reference:arachnids,181; classtype:shellcode-detect; sid:648;
rev:6;)
```

Acima temos uma pequena cadeia de 14 instruções NOPs sendo detectadas pela interface.

Se um pacote é transmitido contendo 14 instruções NOPs, com seu equivalente em hexa, 0x90, o NIDS vai ser capaz de detectar o ataque:

```
[**] [1:648:7] SHELLCODE x86 NOOP [**]
[Priority: 0]
06/23-10:47:09.833341 10.24.0.159:5000 -> 10.24.0.5:32813
TCP TTL:64 TOS:0x0 ID:17322 IpLen:20 DgmLen:588 DF
***AP*** Seq: 0x494F839 Ack: 0xACF02F1E Win: 0x1D18 TcpLen: 32
TCP Options (3) => NOP NOP TS: 300679 917543826
[Xref => arachnids 181]
```

Como se evadir de tal regra?

Existem várias formas, sendo algumas mais comuns, a substituição da instrução NOP por uma equivalente.

Na arquitetura Intel x86, nós temos uma enorme tabela de equivalência de instruções “vazias” que substituem a instrução NOP.

Algumas delas são:

JMP 0x01 – 0xeb 0x01

INC EAX – 0x40

DEC EAX – 0x48

Operação	Valor em Hexadecimal	Representação ASCII
inc %eax	40	@
inc %ecx	41	A
inc %edx	42	B
inc %ebx	43	C
inc %esp	44	D
inc %ebp	45	E
inc %esi	46	F
inc %edi	47	G
dec %eax,	48	H

A lista mais completa pode ser vista em:

<http://cansecwest.com/noplist-v1-1.txt>

Outra possível forma de se evadir é o não uso de NOPs. É possível em determinados casos, encontrar um endereço de retorno “exato”, fazendo com que não haja a necessidade do uso de NOPs. Se um atacante é capaz de fazer depuração no sistema alvo, ele pode obter informações capazes de fornecer o endereço de retorno. Isso é muito comum em ataques envolvendo “Format Strings”.

Ele pode ainda tentar descobrir o endereço exato utilizando um brute force inteligente.

- Regras Contra Shellcode

Uma das formas de tentar detectar um ataque através de NIDS, é a utilização de Regras para tentar detectar shellcodes públicos. A grande maioria dos exploits públicos tentarão executar um shellcode que executa um simples “/bin/sh” em sistemas Unix(Linux), por

exemplo.

Sabendo disso, um NIDS pode ficar a espera da string “/bin/sh” passar em modo binário, detectando assim uma possível tentativa de ataque.

Um exemplo de regra para isto usando o snort segue abaixo:

```
alert ip any any -> any any (msg:"SHELLCODE Linux shellcode";content:"|2f
32 39 3e 2f 43 38|"; reference:arachnids,343;sid:652; rev:9;)
```

Acima vemos a string /bin/sh sendo aguardada como conteúdo de um pacote a espera para “berrar” como uma tentativa de ataque. Um shellcode padrão Mudge/Aleph0ne seria detectado facilmente. Abaixo temos o exemplo de um:

```
char shellcode[] =
"\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b"
"\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"
"\x80\xe8\xdc\xff\xff\xff/bin/sh";
```

No entanto, existem inúmeras formas de se passar por este mecanismo de proteção, algumas delas são:

+ Executar outro comando ou outra shell;

```
char shellcode[] =
"\xeb\x1f\x5e\x89\x76\x0a\x31\xc0\x88\x46\x09\x89\x46\x0e\xb0\x0b\x89"
"\xf3\x8d\x4e\x0a\x8d\x56\x0e\xcd\x80\x31\xdb\x89\xd8\x40\xcd\x80\xe8"
"\xdc\xff\xff\xff/bin/tcsh";
```

+ Não executar uma shell propriamente dita, mas usar read() e execve();

+ Encriptar o Shellcode;

```
"\x31\xc0\xeb\x22\x5b\x8b\x53\x08\x31\x13\x31\x53\x04\x31\xd2\x89"
"\x5c\x24\x08\x89\x54\x24\x0c\xb0\x0b\x8d\x4c\x24\x08\xcd\x80\x31"
"\xdb\x89\xd8\x40\xcd\x80\xe8\xd9\xff\xff\xff"
"\x7a\x37\x3c\x3b\x7a\x26\x3d\x55" /* /bin/sh encriptado */
"\x55\x55\x55\x55"; /* Chave conversora */
```

Logo, da mesma forma que podemos usar um shellcode criptografado para evadir de um

NIDS, nós também podemos usar um “*sistemas de polimorfimo*” para gerar um decrypt(rotina de decriptografia) e uma chave conversora mutável(conceito de polimorfimos de Dark Avenger presente no mundo da escrita de vírus) em um shellcode e conseguir evadir do NIDS.

- Regras Contra Exploits

Muitas vezes um sistema NIDS visa detectar ataques através da exploração específica ou de exploits específicos.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg:"EXPLOIT gobbles SSH exploit attempt"; flow:to_server,established; content:"GOBBLES"; reference:bugtraq,5093; classtype:misc-attack; sid:1812; rev:2;)
```

Para se evadir desta regra basta utilizar um exploit seu ou alterar qualquer pacote inútil que sirva como detecção do ataque. Para quebrar a regra acima, bastaria mudar ou remover a string GOBBLES de qualquer pacote enviado pelo exploit(alterando o source ou binário do exploit).

Ataques de Portscan

- Regras Detectando PortScan(NMAP)

Atuando da mesma forma como detecta ataques de exploração, os sistemas NIDS executam uma checagem de pacotes para analisar ataques de portscan. Vejamos um exemplo de regra para detecção de portscan através do uso da ferramenta mais conhecida de portscan que é o NMAP:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN nmap XMAS"; stateless; flags:FPU,12; reference:arachnids,30; classtype:attempted-recon; sid:1228; rev:3;)
```

Técnicas de Evasão:

- Passive Scanning (p0f);

Passive Scanning é usado para executar ataques de OS/Server Fingerprint e tem sido explorado há mais de 5 anos publicamente. Um atacante pode analisar os pacotes que são recebidos e comparar numa base de dados previamente configurada que será capaz de determinar qual a TCP/Stack está em execução no host alvo.

De forma passiva, o atacante executa uma conexão legítima (finaliza o handshake) e analisa a resposta do host alvo. Desta forma, ele consegue obter dados que serão analisados em sua base de dados interna em busca do OS/Fingerprint.

```
/* Exemplo do uso do p0f -A -C -V */

p0f - passive os fingerprinting utility, version 2.0.4-beta1
(C) M. Zalewski <lcamtuf@disone.cc>, W. Stearns <wstearns@pobox.com>
[+] Signature collision check successful.
p0f: listening (SYN+ACK) on 'eth0', 57 sigs(1 generic), rule: 'all'.
10.24.0.4:80 - Linux recent 2.4 (1) (up: 868 hrs)
  -> 10.24.0.159:32819 (distance 0, link: ethernet/modem)
+++ Exiting on signal 2 +++
[+] Average packet ratio: 6.67 per minute
```

- Idle Scan;

Um Blind Portscan que faz com que o NIDS pense que o ataque está sendo executado através de host Zombie. O ataque se deve através do envio de pacotes spoofados a máquina alvo com o endereço de origem apontando para o host Zombie, o atacante então analisa a incrementação do campo IP_ID na porta 0 no host Zombie.

Os logs que serão gerados no NIDS irá ser os do host zombie.

```
# nmap -P0 -p21,22,25,80 -sI 10.24.0.194 10.24.0.4
Starting nmap 3.48 (http://www.insecure.org/nmap/) at 2004-07-02 12:00 BRT
Idlescan using zombie 10.24.0.194 (10.24.0.194:80); Class: Incremental
Interesting ports on srvce04 (10.24.0.4):
PORT      STATE SERVICE
21/tcp    closed ftp
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
Nmap run completed -- 1 IP address (1 host up) scanned in 4.210 seconds
```

- LD Scanning;

Instala um sniffer na interface, em seguida envia um pacote SYN spoofado para uma máquina alvo e fica aguardando para ver se SYN+ACK é enviado, deste modo, a porta estaria aberta e o IP que será logado no host servidor e no NIDS será o IP Spoofado.

Ataques de SQL Injection/Web Hacking

- Regras Geram muito Falso Positivo;

Pode-se quebrar regras de detecção utilizando *de-sincronização*, passando caracteres inválidos que serão processados pelo NIDS, mas não serão processados pela aplicação:

Um exemplo básico consiste em enviar a seguinte sequência de pacotes:

GET /cgi-b – (Pacote Válido)

NASH - (Pacote Inválido)

in – (Pacote Válido)

O NIDS interpreta como sendo “GET /cgi-NASHin” já o sistema operacional interpreta “GET /cgi-bin”.

Podemos ainda evadir as regras usando validações de entrada(*input validation*).

Se algum NIDS utiliza, por exemplo, uma regra para detectar ataques de SQL Injection que usam a comando “OR 1=1”, um atacante pode evadir esta regra através de uma validação de entrada que produz o mesmo resulta, por exemplo: “OR 2=2”. Existem várias formas de se obscurecer uma URL.

Podemos “*obscurecer uma url*” passando por uma checagem de regras do NIDS. Um exemplo clássico pode ser visto abaixo:

URL Normal: <http://www.pc-help.org/obscure.htm>

URL Obscurecida: [http://!\\$^&*\(\) +`-= { } | | : ; @ www.pc-help.org/obscure.htm](http://!$^&*() +`-= { } | | : ; @ www.pc-help.org/obscure.htm)

Dependendo do navegador que você usa, o endereço acima pode o levar a caminhas inesperados...). Mas lembre-se que estamos atacando “um servidor” e não um navegador.

Existem vários meios para obscurecer uma URL, consulte o site acima para ver alguns.

Uma ferramenta que executa muitos desses ataques se chama Whisker e pode ser obtida no seguinte link: <http://www.wiretrip.net/rfp>.

Ataques de Brute Force

- Sistema de Timer Contra Brute Force

Um sistema pode ser capaz de armazenar em buffer os ataques de brute force em contas específicas, por exemplo, as contas capturadas através de emails(smtp expn, vrfy). Um atacante pode ir tentando senha por senha até descobrir uma conta válida. Supondo que

o NIDS atue de forma ativa, ele poderia “dropar” a conexão após X tentativas inválidas.

Possíveis, ataques incluem o conceito de Slow scan. Onde se pode tentar uma tentativa agora e outra depois de horas. No entanto, este ataque pode ainda não ser efetivo, pois a demora em ataques de brute force pode chamar a atenção de forma demasiadamente grande!

De modo que, surge a necessidade de atacar utilizando ferramentas distribuídas. Supondo um NIDS que bloqueia a conta por 5 minutos após 5 tentativas, um atacante poderia trabalhar alternando máquinas, uma máquina A testa 5, depois de 1 minuto, outra B testa 5, depois de 1 minuto outra máquina C testa mais 5 e quando a quarentena de 5 minutos terminar para a máquina A, ela volta a testar 5. Dependendo do daemon, é ainda possível atacar utilizando “thread”, com múltiplas conexões oriundas de Ips diferentes ao mesmo tempo.

Backdoor e Trojan Horses

- Regras Detectando Inúmeras Backdoors e Trojan Horses

A maioria dos NIDS checam por pacotes vindo de fora para dentro da rede. E conexões legítimas são muitas vezes passadas despercebidas.

Os ataques de tunelamento são eficazes como backdoor e trojan horse. Atuando utilizando o conceito de “Connect-Back”, é possível a um atacante criar um túnel que não chame a atenção do sistema alvo.

Um exemplo real, é a criação de um túnel http com dados sendo criptografados e enviados via parâmetro GET de um servidor. Conceito da Little Crow.

Existem ainda os “*convert channel*”, que atuam em cima de um protocolo ou especificações do TCP/IP capazes de ser interpretado pelo host alvo, mas descartado pelo NIDS. Um exemplo é uma backdoor que envia pacotes criptografados com ISN inválidos.

Pegando um gancho no conceito acima, backdoors estilo “*bindshell criptografadas*” também podem ser usadas para se evadir de um software NIDS.

Temos visto o uso do SSH para tunelamento de vários protocolos(telnet, ftp, dns), logo este tipo de backdoor consegue ser bastante evasiva.

Denial Of Service

Duas classes de Denial of Service se sobressaem, são elas:

- Alert Flood:

Stick - <http://www.eurocompton.net/stick/projects8.html>

Snot - <ftp://ftp.st.ryukoku.ac.jp/pub/security/tool/snot>

Atua gerando muitas entradas falsas, com pacotes spoofados no host alvo, dificultando qualquer posterior análise forense.

- DoS na Aplicação:

Integer Overflow em Stream4 no Snort

Atua derrubando o serviço do NIDS, que executa como um daemon normal e pode apresentar problemas no software.

Outros Problemas em NIDS

- **Redes Switched (Problemas com Sniffing)**

Para que um NIDS atue de forma correta, ele precisa monitorar eventos em toda a Rede. Alguns NIDS atuam com sensores espalhados pela rede fazendo uma espécie de comunicação “cliente e servidor”, onde os sensores são espalhados pelas máquinas da rede e enviam logs para um servidor NIDS Central, que gera uma base de dados de logs.

No entanto, a maioria dos NIDS atua apenas com um servidor central escutando numa interface em modo promíscuo, todos os dados transitando pela rede. Deste modo, ele analisa os pacotes transitando pelas estações em busca de dados suspeitos.

Se um NIDS atua com sensores espalhados pelas estações e um atacante obtém acesso a uma estação, ele pode desde derrubar o sensor até mesmo utilizar este sensor para ludibriar o sistema de IDS, e pode ainda utiliza o sensor para interagir diretamente com o sistema NIDS.

Se um NIDS trabalha com o modelo centralizado, uma rede segmentada pode representar um problema, pois não bastaria apenas setar a interface em modo promíscuo para obtenção dos pacotes transitando pela rede. Outros recursos então seriam necessário, fazendo com que, em alguns casos, seja necessário a mudança na topologia da rede.

- **Redes de Alta-Transmissão**

Muitas redes transmitem dados em altíssima velocidade. A grande maioria dos sistemas NIDS trabalha coletando e manuseando dados a no máximo 100 Mbps. Alguns IDSs não possuem suporte a tecnologias como ATM e GigaBit Ethernet, de modo que, muitas vezes se faz necessário alterações na Topologia de uma Rede para a utilização de um NIDS pela mesma.

Em Redes de Alta-Transmissão, a análise de regras pode exigir um processamento considerável, e consequentemente, máquinas com poder de processamento grande.

Algumas soluções tem sido estudadas. Uma delas diz respeito a utilização de Massivo Processamento Paralelo(MPP) para trabalhar na leitura das regras em redes de grande porte, onde vários sistemas trabalham em cima de “Clusters”, processando dados paralelamente visando conseguir detectar os ataques com o menor número de falsos positivos possível.

No entanto, os custos envolvidos nestes projetos são consideráveis e os problemas envolvendo falso positivos são reais.

- **Falhas no Software NIDS (libpcap, libnids, etc)**

Os NIDS nada mais são que softwares atuando como sniffers e tratando pacotes transitando em uma interface de Rede. Sendo assim, problemas inerentes a qualquer daemon/servidor, também podem estar presentes nos softwares NIDS. Exemplos públicos e atuais existem aos montes. Alguns que destaco, são os seguintes:

* **Snort TCP Packet Reassembly Integer Overflow Vulnerability**

(<http://www.securityfocus.com/bid/7178>);

Há aproximadamente 1 ano atrás, vemos uma condição de buffer overflow existente no Snort capaz de permitir execução remota de comandos na máquina possuidora do NIDS. A falha estava presente no pré-processador stream4, responsável pelo reassembly de pacotes TCP fragmentados.

* **Internet Security Systems Protocol Analysis Module SMB Parsing Heap Overflow Vulnerability**

(<http://www.securityfocus.com/bid/9752>)

Um Módulo que trata pacotes SMB, existente em diversos produtos da ISS, contém uma falha de segurança que pode permitir a um atacante executar comandos arbitrários num sistema NIDS. Este problema está presente nos produtos RealSecure, Proventia, dentre outros da ISS.

Deve-se notar que esta falha foi reportada em Fevereiro deste ano(2004).

*** Libnids TCP Packet Reassembly Memory Corruption Vulnerability**

(<http://www.securityfocus.com/bid/8905>)

LIBNIDS é uma biblioteca de alto-nível utilizada para construir sistemas NIDS de forma automatizada. No dia 15 de outubro de 2003 foi tornada pública uma condição de buffer overflows num recurso de manuseio de pacotes TCP fragmentados. Esta falha permitia a execução de comandos remotos numa máquina com a LibNIDS em execução.

- Serviços Criptografados (VPN)

Para que a análise de pacotes por parte de um NIDS seja efetiva ele precisa no mínimo conhecer de forma limpa(entender) os dados que estão transitando na interface. Se um serviço está sendo criptografado, se torna inviável para um NIDS analisar os pacotes e consequentemente muitos ataques podem passar despercebidos.

Com o aumento no número de sistemas e serviços utilizando o conceito de VPN(Virtual Private Network), se torna mais comum este tipo de problema.

Ferramentas Públicas

Whisker (Web Evasion) – <http://www.wiretrip.net/rfp/>

NIDSBench (Fragrouter / TCPReplay) –
<http://packetstorm.widexs.nl/UNIX/IDS/nidsbench/nidsbench.html>

IDSWakeUp - <http://www.hsc.fr/ressources/outils/idswakeup/>

Congestant – <http://www.phrack.org/> - 54

Ftester - <http://sourceforge.net/projects/ftester/>

ISIC - <http://www.packetfactory.net/projects/ISIC/>

Mendax - <http://adam.kaist.ac.kr/~bugsy/mendax.html>

Siden - <http://siden.sourceforge.net/>

O Futuro dos NIDS

Relatório da Gartner (2003)

Em um relatório intitulado “Intrusion Detection Is Dead - Long Live Intrusion Prevention”, a empresa de consultoria Gartner recomendou a adoção de sistemas de Statefull Firewalls ao invés da tecnologia de IDS, alertando que a mesma já estaria ultrapassada.

http://www.giac.org/practical/GSEC/Tim_Wickham_GSEC.pdf

Sistemas de IDS com Data Mining e Spectrum Analysis

Existe uma forte tendência de agregar sistemas NIDS com soluções locais, como TPE(Trusted Path Execution), systrace e etc. Para que os NIDS funcionem de maneira efetiva, se faz necessário uma análise mais apurada dos dados, manuseando recursos e base de dados e análise spectral.

Sistemas de Inteligência Artificial

Alguns sistemas NIDS já estão sendo criados utilizando conceitos de redes neurais, onde se analisa os pacotes com base em tráfego normal e tráfego anômalo. Os sistemas de NIDS com inteligência artificial detectam tráfego anômalo e procuram ver se um ataque está em execução. Muita pesquisa tem sido desenvolvida nesta área, com métodos de objetivação e sistemas especialistas sendo criados.

Fraquezas Conceituais Futuras

No caso dos sistemas de IDS com Mineração de Dados e Análise Spectral, os ataques de evasão ainda podem ser utilizados, onde o atacante pode criar meios de gerar “Falsos Negativos”, utilizando metamorfismo e etc.

No caso de Sistemas com Inteligência Artificial, os modelos existentes aprendem em cima de regras, onde um atacante pode descobrir como ocorre o aprendizado(seja via motor de

inferência, etc) e criar ataques que levem o NIDS a pensar que se trata de atitudes normais do sistema.

Conclusão

Um sistema NIDS pode representar um considerável aumento da segurança contra ataques padrões ou atacantes com conhecimentos básicos. A situação não é a mesma quando atacantes avançados ou grupos de atacantes avançados estão por trás das tentativas.

De modo que, as tecnologias atuais de NIDS podem não representar grandes empecilhos contra atacantes avançados, isso inclui NIDS comerciais com custo elevado de manutenção. O futuro pode exigir maior empenho dos atacantes, mas as tecnologias já estão nascendo com fraquezas conceituais.

Um sistema NIDS jamais pode ser considerado como uma solução efetiva aos problemas de segurança. E o seu uso como complemento jamais deve ser primário (exemplo da atualização da Base de Dados do NIDS antes de um patche num software vulnerável).

Outro fator importante que consideramos é o custo de manutenção deste tipo de solução, onde existe a necessidade de atualização permanente, e no caso de empresas com regras (ou serviços) específicas o acompanhamento da atualização e inserção de novas regras por um Analistas de Segurança.

Nenhum software, mesmo um sistema especialista, é capaz de substituir um especialista humano, de modo que, as empresas e a comunidade de segurança precisam debater a questão da conscientização e elevação dos conhecimentos técnicos dos envolvidos em gerenciar a segurança.

Um Cordial Abraço a Todos,

Nash Leon.

Créditos

Imagens usadas na apresentação:

Imagem da Apresentação – Google - revir.com.br/desktop/gotico/gotico05.jpg

Imagem seguinte – Yahoo – www.katemary.blogspot.com.br

Gotica4 - [http://luanegra.blogs.sapo.pt/arquivo/\(83\).jpg](http://luanegra.blogs.sapo.pt/arquivo/(83).jpg)

Cemiterio1 - <http://luanegra.blogs.sapo.pt/arquivo/graves.gif>

Gotica013 - <http://planeta.terra.com.br/lazer/rankincs/gotica013.jpg>

Gotica22 - <http://planeta.terra.com.br/lazer/rankincs/gotica022.jpg>

Gotica030 - <http://planeta.terra.com.br/lazer/rankincs/gotica030.jpg>

Gotica032 - <http://planeta.terra.com.br/lazer/rankincs/gotica032.jpg>
Gotica039 - <http://planeta.terra.com.br/lazer/rankincs/gotica039.jpg>
Gotica041 - <http://planeta.terra.com.br/lazer/rankincs/gotica041.jpg>
Gotica064 - <http://planeta.terra.com.br/lazer/rankincs/gotica064.jpg>
Gotica142 - <http://planeta.terra.com.br/lazer/rankincs/gotica142.jpg>
Gotica182 - <http://planeta.terra.com.br/lazer/rankincs/gotica183.jpg>
Coracao - http://www.fotolog.net/sektors/?photo_id=7744962
ArteWork.jpg - <http://digitalart.org/artwork.php?ID=34171>
Artework2.jpg - <http://digitalart.org/artwork.php?ID=33408>
ArtWork3.jpg - <http://digitalart.org/artwork.php?ID=26230>
Artwork4.jpg - <http://digitalart.org/artwork.php?ID=35511>
ArtWork5.jpg - <http://digitalart.org/artwork.php?ID=32048>
ArtWork6.jpg - <http://digitalart.org/artwork.php?ID=30604>
Sol1.jpg - <http://www.fotoblog.com/images/uploads/mimetic03.jpg>
Dark1.jpg - <http://www.fotolog.net/danny/>
Dark2.jpg - <http://www.fotolog.net/blackphoenix>
Dark3.jpg - <http://www.fotolog.net/blackphoenix>
Hacker1.gif - <http://www.q1labs.com/newsletter/nlimages/hacker.gif>
Hacker2.gif - <http://www.itportal.it/special/sicurezza/crearevirus/hacker.jpg>
Seta1.gif - <http://si.porto.ucp.pt/alunos-mestrado/suzana/design/seta.gif>
Seta2.gif - <http://www.classeatur.com.br/imagens/seta.gif>
Matrix1.jpg - http://www.icoaraci.com.br/papel_parede/imagens/pp_da_010p.jpg
Hacker3.jpg - <http://rhein-zeitung.de/on/03/08/22/internet/hacker.jpg>
Rede1.gif - <http://www.networkdesigners.com.br/Artigos/nat/nat-rede2.gif>
Rede2.gif - <http://www.epub.org.br/informed/hall02.gif>
Espada.jpg - <http://www.dreamers.com/manuscritos/imagenes/papeltap/800x600/espada.JPG>
Gotico5.jpg - <http://www.gothmetal.net/ezimagecatalogue/catalogue/variations/714-200x200.jpg>

Outros Links de Referência:

<http://www.nss.co.uk/>
<http://www.securityfocus.com/>
<http://www.insecure.org/>
<http://www.packetstormsecurity.nl/>