Referências Bibliográficas

[1] "Enterprise Security Product Markets". DataMonitor, Available via URL: http://www.datamonitor.com, Reference Code DMTC0913, 2003.

[2] "Estatísticas CERT/CC", URL : http://www.cert.org .

[3] Howard, John D. An Analysis of Security Incidents on the Internet : 1989 – 1995. Pittsburgh, 1997. Tese de Doutorado.

[4] Spafford. Eugene H. The Internet Worm Program : An Analysis. In ACM Computer Communication Review; 19 (1), 17-57, Janeiro de 1989.

[5] Spafford. Eugene H. The Internet Worm Incident. ESEC' 89 : 2nd European Software Engineering Conference. 1989.

[6] Moore. David, Shannon. Colleen, Brown. Jeffery. Code-Red : A case study on the spread and victims of an Internet Worm. In Proceedings of Internet Measurement Workshop. 2002.

[7] NBSO – NIC BR Security Office. URL : http://www.nbso.nic.br.

[8] R. Heady. Et al. The Architecture of a Network Level Intrusion Detection System. Technical Report, Department of Computer Science, University of New Mexico, August 1990.

[9] Denning, Dorothy. "An intrusion detection model". IEEE Transactions on Software Engineering, Vol SE 13 No 2, February 1987.

[10] S. Axelsson. Research in Intrusion Detection Systems: A Survey. Technical Report. 1999.

[11] Anderson P. James. "Computer Security threat monitoring and surveillance". Technical Report Contract 79F26400, 1980.

[12] Cunningham R. Lippmann R - Improving Intrusion Detection performance using Keyword selection and Neural Networks R - MIT Lincoln University (http://www.ll.mit.edu/IST/pubs.html), 1999.

[13] Rhodes, B. Mahaffey, J. Cannady, J. Multiple Self-Organizing Maps for Intrusion Detection. Proceedings of the 23rd National Information Systems Security Conference. October 2000.

[14] Roesch, Marty. SNORT. URL: http://www.snort.org.

[15] Porras, P. A, Kermerer, R.A. State Transition Analysis : A Rule-based Intrusion Detection Approach. In Proceedings of the Eight Annual Computer Security Applications Conference. December 1992.

[16] Ilgun, Korah. USTAT: A Real Time Intrusion Detection System for Unix. In Proceedings of the 1993 IEEE Symposium on Security and Privacy.. 1993.

[17] IETF: Internet Enginnering Task Force. URL : http://www.ietf.org

[18] IETF Intrusion Detection Exchange Format Working Group. URL : http://www.ietf.org/html.charters/idwg-charter.html

[19] Security Focus. URL : http://www.securityfocus.com.

[20] PacketStorm Security. URL : http://www.packetstormsecurity.com.

[21] BugTraq Mailing List Archive. URL : http://msg.securepoint.com/bugtraq.

[22] Weber, Daniel. A Taxonomy of Computer Intrusions. Master's Thesis. Massachussets Institute of Technology, Cambridge, MA. 1998.

[23] C. Schuba, Et al. Analysis of a denial of service attack on TCP. In Proceedings of the 1997 IEEE Symposium on Security and Privacy, pages 208--223. IEEE Computer Society Press, May 1997.

[24] Allard. F.; Fuchs, J. Artificial Intelligence - The State of the Art. ESA's Technology Programme Quartely. Vol.3. N.3. Sep. 1993.

[25] White, D. A. Sofge, D. A. Artificial Neural Networks in Manufacturing and Process Control, in Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches., Van Nostrand Reinhold, Florence, KY.

[26] Bishop, C. M.. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, UK. 1995.

[27] Haykin, S. Neural Networks : A comprehensive foundation. Prentice Hall. 1999.

[28] M. Minsky and S. Papert. *Perceptrons*. MIT Press, 1969.

[29] Velasco, Marley Maria. Apostila de Redes Neurais Artificiais. Puc-Rio.

[30] Stone,M. Cross-validation : A review. Mathematische Operationsforschung Statitischen, Serie Statitics, vol.9, pp. 127-139. 1978.

[31] Lawrence Berkeley National Laboratory. tcpdump. URL : http://www.tcpdump.org.

[32] KDDCUP 1999 DataSet Repository. URL : http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[33] Osterman, S. tcptrace. URL : http://www.tcptrace.org

[34] Microsoft Corp. URL : http://www.microsoft.com/sql

[35] R. Kohavi, and F. Provost. Glossary of Terms, Editorial for the special issue on Application of Machine Learning and the Knowledge Discovery Process, Vol 30, No 2/3, Feb/March 1998.

[36] Pfahringer B.:Winning the KDD99 Classification Cup: Bagged Boosting,SIGKDD explorations, 1(2), 65-66, 2000

[37] Bugtraq Archives (e-mail regarding Apache vulnerability). URL: http://www.geek-girl.com/bugtraq/1998_3/0442.html. August 7, 1998.

[38] CERT Incident Note. URL: http://www.cert.org/incident_notes/IN-98.02.html. July 2, 1998.

[39] NMAP Homepage. http://www.insecure.org/nmap/index.html. 1998

[40] SAINT. URL: http://www.saintcorporation.com

[41] Farmer, Dan. Wenema, Vietse. “Improving the security of you site by breaking into it”. 1993.

[42] CERT Advisory CA-93.10. http://www.cert.org/ftp/cert_advisories/CA-93%3a10.anonymous.FTP.activity. July 14, 1993.

[43] CERT Advisory CA-97.09. http://www.cert.org/ftp/cert_advisories/CA-97.09.imap_pop. April 7, 1997.

[44] Impack binaries on Rootshell.com. http://www.rootshell.com/archive-j457nxiqi3gq59dv/199804/impack103.tar.gz.html. April 13, 1998.

[45] CERT Advisory CA-98.05. http://www.cert.org/ftp/cert_advisories/CA-98.05.bind_problems. April 8, 1998.

[46] CERT Advisory CA-96.06. http://www.cert.org/ftp/cert_advisories/CA-96.06.cgi_example_code. March 20, 1996.

[47] CERT Advisory CA-97.05. http://www.cert.org/ftp/cert_advisories/CA-97.05.sendmail. January 28, 1997.

[48] Sun Microsystems Security Bulletin: #00138. http://sunsolve.Sun.com/pub-cgi/us/sec2html?secbull/138. 17 April, 1997.

[49] Sun Microsystems Security Bulletin: #00140. http://sunsolve.Sun.com/pub-cgi/us/sec2html?secbull/140. 14 May, 1997.

[50] CERT Advisory CA-96.12. http://www.cert.org/ftp/cert_advisories/CA-96.12.suidperl_vul. June 26, 1996.

[51] CERT Advisory CA-95.09. http://www.cert.org/ftp/cert_advisories/CA-95.a09.Solaris-ps.vul. August 20, 1995.

Apêndice A
Padrões de ataque – Negação de Serviço

Existem 11 diferentes padrões de ataque e intrusão da classe negação de serviço na base de dados do Kddcup 1999 – ver tabela 3.5 - utilizada neste trabalho. Uma breve descrição da operação de cada um destes ataques será apresentada neste apêndice.

A.1 Apache2 R-a-Deny (Temporary/Administrative)

Ataque Apache2 [37] tem como alvo servidores Web (servidores que interagem com navegadores Internet utilizando protocolo HTTP) Apache. O ataque consiste no envio de requisições iniciais do protocolo HTTP contendo um alto número de cabeçalhos HTTP. Enquanto uma requisição HTTP normal possui no máximo 20 cabeçalhos HTTP, uma requisição gerada para um ataque Apache2 possui, tipicamente, mais de 1000 cabeçalhos HTTP. O conteúdo de cada cabeçalho não é importante. Servidores Apache recebendo estas solicitações anômalas aumentam seu consumo de recursos computacionais como processador e memória reduzindo drasticamente o seu desempenho e tempo de resposta para usuários legítimos. Em alguns casos, o servidor vítima deste ataque pode se tornar completamente inoperante, parando de responder a quaisquer solicitações de usuários.

A.2 Back R-a-Deny (Temporary)

Também direcionado contra servidores Web Apache, o ataque Back consiste em envio de solicitação HTTP contendo URL com um alto número – tipicamente mais de 100 - de caracteres “/” (barra invertida). Ao receber uma solicitação com este URL anômalo, servidores Web Apache2 aumentam sua utilização de CPU afetando o desempenho geral do sistema vítima. O processamento usualmente é normalizado com o fim do ataque.

A.3 Land R-b-Deny (Administrative)

O padrão de ataque Land tem como alvo determinadas implementações do conjunto de protocolos TCP/IP. O único sistema vulnerável a este padrão de ataque existente no ambiente montado pelo projeto DARPA/MIT era o sistema operacional SunOS 4.1. Land consiste no envio de um pacote TCP-Syn (usado para iniciar a negociação de inicio de conexão TCP) com endereço de origem igual ao endereço de destino. Ao receber este pacote o sistema operacional SunOS 4.1 ficava completamente inoperante. Para recuperar este sistema é necessário desligar e ligar o computador.

É um padrão de ataque facilmente detectado por sistemas de detecção de intrusão uma vez que um pacote TCP/IP com endereços de origem e destino idênticos jamais deveria trafegar em redes TCP/IP.

A.4 MailBomb R-a-Deny (Administrative)

MailBomb consiste no envio de milhares de mensagens de e-mail para um único usuário com o objetivo de sobrecarregar o processamento de servidor de e-mail (baseados no protocolo SMTP). Implementações típicas deste ataque enviam 10000 mensagens de 1Kbyte (10 Mbytes de dados) para um único usuário continuamente. O efeito deste ataque esta reduzido a uma sobrecarga de processamento no servidor SMTP e ao recebimento de milhares de mensagens inúteis na caixa postal do usuário.

Identificação de um ataque Mailbomb é relativamente subjetiva e depende de uma definição de quantas mensagens de um mesmo remetente para um mesmo usuário em um determinado intervalo de tempo podem ser consideradas normais para cada organização.

A.5 SynFlood R-a-Deny (Temporary)

Ataques SynFlood – também conhecidos como Neptune devido ao nome dado ao primeiro programa que implementou esta técnica de intrusão - abusam de característica normal e padrão do protocolo TCP. Antes de qualquer comunicação baseada no protocolo TCP começar é necessário o estabelecimento de uma conexão virtual entre as partes envolvidas na

comunicação. Esta “conexão” é estabelecida durante um “handshake” ou negociação inicial. A figura A.1 ilustra este processo.

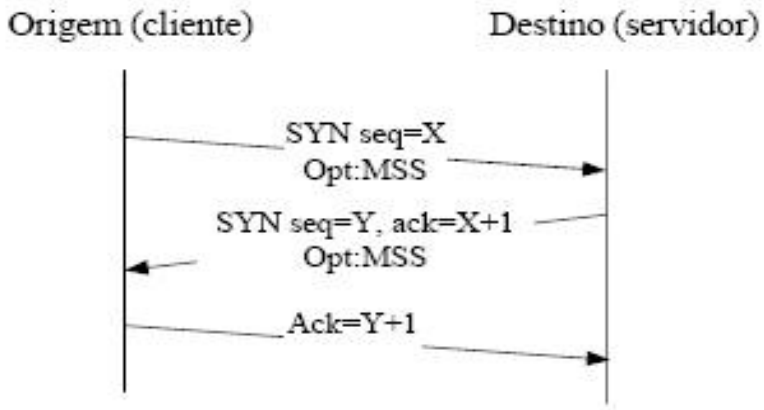

Figura A.1 : Conexão inicial TCP

O computador destino de uma negociação de uma nova conexão TCP aloca memória para a nova conexão assim que recebe o primeiro pacote TCP-Syn do cliente. Se não for recebido o 3º pacote do processo – TCP-Ack – após um determinado período o computador destino assume que algum problema ocorreu e libera os recursos pré-alocados. O ataque SynFlood explora esta alocação inicial de recursos enviando um enorme número de pacotes TCP-Syn ao servidor sem, entretanto, completar a conexão. O servidor alocará recursos para conexões falsas comprometendo recursos que poderiam estar disponíveis para conexões legítimas e, em casos extremos, ficando completamente inoperante devido a sobrecarga de solicitações. Algumas implementações TCP/IP quando vítimas de um ataque Synflood acabam incorrendo em uma situação de exceção e paralisando completamente o funcionado do sistema operacional.

Um sistema de detecção de intrusão para redes pode detectar este ataque pela monitoração e comparação do número de pacotes TCP-Syn enviados para determinado computador contra um determinado limiar máximo. Um sistema de detecção de intrusão para host também pode ser eficaz em

detectar este tipo de ataque através da monitoração do crescimento das áreas de memória reservadas pelo sistema operacional para estruturas de controle de conexões TCP.

A.6 PingofDeath R-b-Deny (Temporary)

Ataque negação de serviço que consiste no envio de pacotes ICMP-Echo (gerados normalmente através do utilitário “ping” para diagnóstico) com tamanhos superiores a 64000 bytes. Como pacotes ICMP-Echo são normalmente bem inferiores a este valor diversos sistemas operacionais e implementações de TCP/IP disponíveis não conseguiam tratar esta anomalia e reagiam de forma inesperada – as reações mais comuns eram reinicialização do computador ou total paralisação de todas as operações.

Sistemas de detecção de intrusão podem facilmente detectar este padrão de ataque monitorando o tamanho de pacotes ICMP-Echo.

A.7 ProcessTable R-a-Deny (Temporary)

Ataque criado durante o projeto MIT/DARPA de avaliação de SDIs, o ataque ProcessTable tem como alvo diversas variantes do sistema operacional Unix. O princípio básico para o funcionamento deste ataque esta relacionado a como sistemas operacionais Unix lidam com conexões e solicitações recebidas pela rede. Tipicamente um novo processo é carregado em memória para lidar com a nova conexão e/ou requisição recebida. Para usuários regulares o sistema operacional limita a quantidade de processos que cada usuário pode carregar em memória. Este limite não é feito, entretanto, para o usuário “root” – usuário com poder de administrador – que normalmente é o usuário utilizado por processos associados a funções de rede do sistema operacional. Um sistema vítima deste ataque tem sua tabela interna de controle de processos ativos sobrecarregada impedindo a ativação de novos processos legítimos.

A.8 Smurf R-a-Deny (Temporary)

Em um ataque Smurf o atacante envia pacotes ICMP-Echo com endereço de destino de “broadcast” – todas as máquinas em uma determinada rede – e endereço de origem igual ao do sistema alvo do ataque. O sistema vítima terá uma sobrecarga ao receber milhares de respostas – pacotes ICMP-Echo-Reply – de solicitações que ele não realizou. A figura A.2 ilustra este processo.

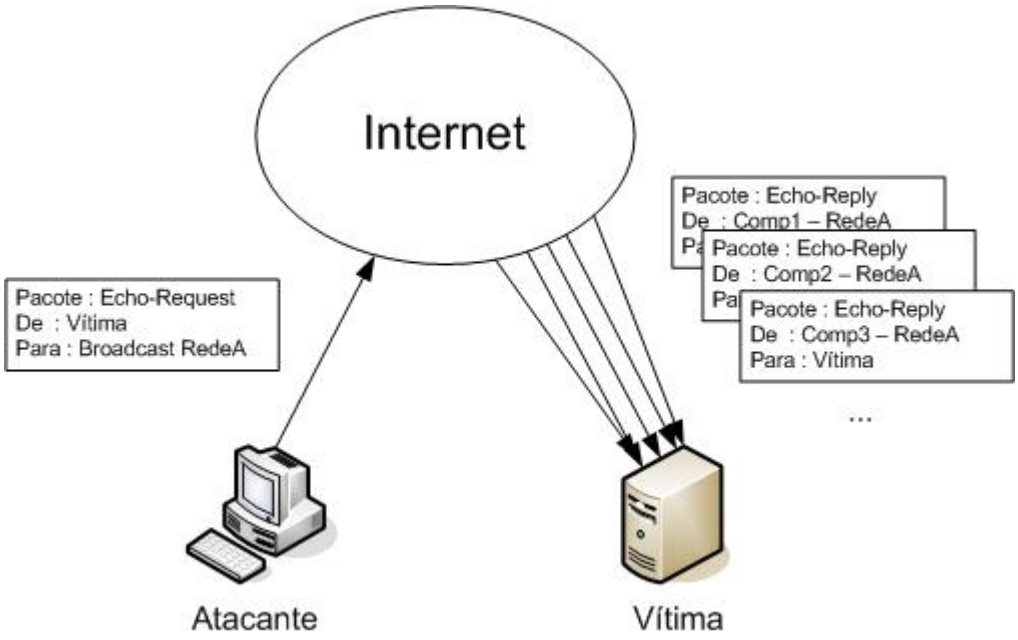

Figura A.2 : Esquema Ataque Smurf

Um sistema de detecção de intrusão pode facilmente detectar um ataque Smurf observando a quantidade de pacotes Echo-Reply recebidos em relação a quantidade de pacotes Echo-Request enviados. Em uma situação ideal normal estes valores devem ser iguais ou bem próximos.

A.9 SyslogD R-b-Deny (Administrative)

Ataque negação de serviço que tem como alvo o serviço Syslog do sistema operacional Solaris. Syslog é o serviço de registro de eventos e auditoria da maioria dos sistemas operacionais Unix. Quando o serviço Syslog recebe um

registro de evento para ser armazenado de um computador remoto ele tenta resolver qual nome DNS esta associado ao endereço IP do computador remoto. Se esta resolução não for possível uma falha no Syslog causa sua indisponibilidade. O serviço só restaurará sua operação normal após intervenção de um administrador do sistema. Este é um ataque que explora uma falha ou "bug" em um serviço de rede. Esta falha ou "bug" esta relacionada ao não tratamento de situações excepcionais ou não previstas em condições normais.

A.10 TearDrop R-a-Deny (Temporary)

Ataque que explora falha no tratamento de pacotes fragmentados em diversas implementações de TCP/IP em vários sistemas operacionais. Um pacote TCP/IP pode encontrar um enlace entre a origem e destino que não suporte trafegar pacotes maiores do que determinado tamanho – parâmetro conhecido como MTU ou "Maximum Transmission Unit". Quando isto ocorre este pacote é fragmentado em um ou mais pacotes. É função do destinatário do pacote realizar a operação de reagrupar estes fragmentos no pacote original. Algumas implementações de TCP/IP não conseguiam realizar este reagrupamento se houvesse alguma sobreposição de dados entre os diversos fragmentos recebidos. Estas implementações causavam a reinicialização do sistema operacional se uma situação não esperada como esta fosse encontrada.

A.11 UDPStorm R-a-Deny (Administrative)

Ataque de negação de serviço que causa congestionamento e alterações no desempenho de redes TCP/IP. Alguns serviços UDP (como "chargen", "echo" e outros) geram resposta automática sem qualquer verificação de origem e autenticidade da solicitação original recebida. Um atacante que envie um pacote com endereço de origem alterado para o da vítima e envie para um serviço desta classe pode gerar um processo repetitivo de comunicação entre o prestador do serviço e a máquina alvo. A figura A.3 ilustra um exemplo de um ataque UDPStorm utilizando o serviço Echo (porta UDP número 8).

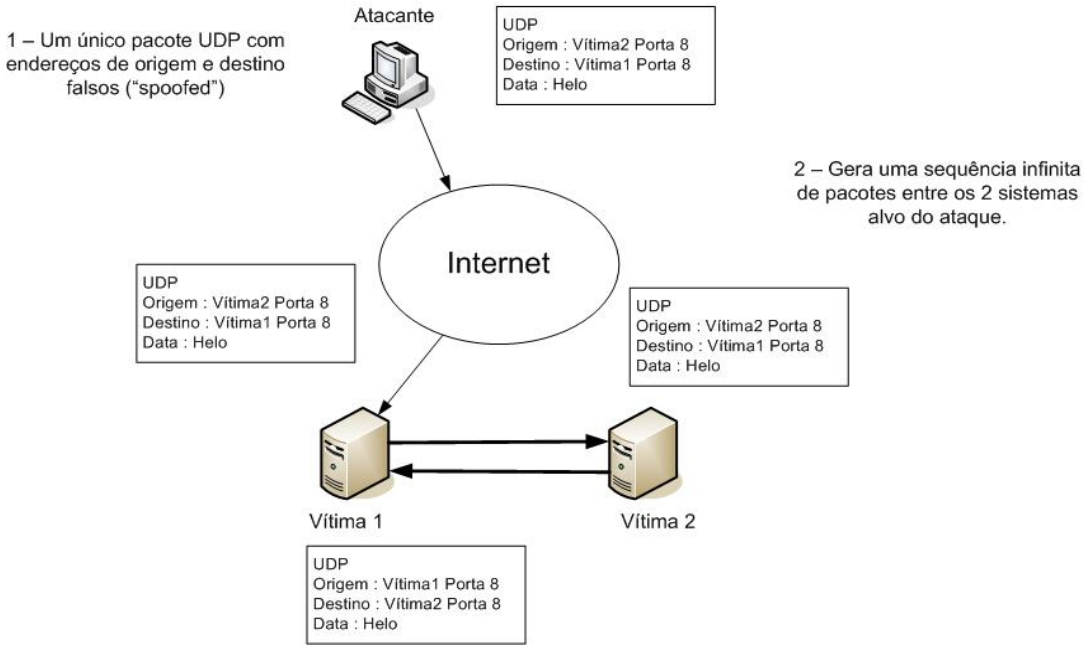

Figura A.3: Exemplo de ataque UDPStorm

Apêndice B
Padrões de ataque – Reconhecimento

Cinco padrões de ataques da classe Reconhecimento ou “Probing” estão presentes na base de dados MIT/KddCup 1999. São eles :

B.1 IPSweep R-a-Probe (Machines)

IPSweep é um ataque utilizado na etapa de reconhecimento e seleção de alvos vulneráveis. Consiste em uma varredura ou busca por computadores e sistemas acessíveis e que possa ser atacados por outros métodos em uma próxima etapa. IPSweep utilizado neste trabalho envia pacotes ICMP-Echo (utilitário Ping) para todos os endereços de uma determinada rede e constrói uma relação de respostas recebidas.

Um sistema de detecção de intrusão que monitore uma seqüência de pacotes ICMP-Echo para endereços de destino em uma faixa continua e vindos de um mesmo endereço de origem será capaz de detectar este padrão.

B.2 MScan R-a-Probe (Known Vulnerability)

MScan [38] é um ataque de reconhecimento que utiliza consulta a servidores DNS e realiza uma busca nos endereços IP encontrados procurando por vulnerabilidades específicas. Com código fonte disponível na Internet, esta ferramenta de ataque pode ser customizada para encontrar diversas vulnerabilidades em potenciais nas máquinas selecionadas através do DNS. Para o projeto MIT/DARPA foram realizados ataques MScan para todos os computadores pertencentes ao domínio eyrie.af.mil procurando pelas seguintes vulnerabilidades : statd, imap,pop, computadores IRIX com contas de usuários sem senha, bind, diversas vulnerabilidades cgi-bin em serviços Web, NFS e serviços X.

B.3 Nmap R-a-Probe (Services)

Nmap [39] é uma ferramenta de varredura e reconhecimento capaz de realizar busca por computadores, serviços e vulnerabilidades utilizando diversos mecanismos. Sua principal função é determinar quais portas TCP ou UDP estão ativas em determinado computador, consequentemente, determinando quais serviços estão ativos. A ferramenta permite ainda a identificação exata (“”fingerprinting”) de versão dos softwares que implementam os serviços descobertos. Esta informação é útil para um atacante determinar quais vulnerabilidades podem ser exploradas em uma 2ª etapa do ataque. Um aspecto da ferramenta Nmap que pode dificultar sua detecção é a capacidade de configurar diversos aspectos relativos a velocidade e periodicidade em que as varreduras são realizadas.

B.4 Saint R-a-Probe (Known Vulnerabilities)

Saint [40] – “Security Administrator Integrated Network Tools” – é uma ferramenta desenvolvida para administradores de redes, sistemas e profissionais de segurança da informação. Esta ferramenta tem como objetivo levantar o maior número de informações sobre serviços ativos em computadores remotos. Apesar de não ter sido desenvolvida com intenção de ser utilizada para ataques e intrusões, a enorme quantidade de informações levantadas podem ser úteis para um invasor/atacante.

B.5 Satan R-a-Probe (Known Vulnerabilities)

Satan [41] – “Security Administrator Tool for Analyzing Networks” – pode ser considerado o predecessor da ferramenta Saint descrita anteriormente. O princípio das duas ferramentas é essencialmente o mesmo, sendo que a diferença entre elas reside apenas nos serviços e vulnerabilidades pesquisados.

Um sistema de detecção de intrusão de rede pode detectar facilmente uma varredura utilizando o Satan devido ao perfil regular de tráfego gerado pela ferramenta.

Apêndice C
Padrões de ataque – Remoto para local

Padrões de ataque da categoria “Remoto para local” são extremamente perigosos pois normalmente permitem que um atacante com apenas a capacidade de enviar pacotes TCP/IP para a vítima, mas sem nenhum outro tipo de acesso, obtenha acesso não autorizado como um usuário legítimo. 9 padrões desta categoria estão presentes na base de dados. São eles :

C.1 Dictionary R-a-U

Padrão de ataque e intrusão clássico em que um atacante tenta obter acesso não autorizado a determinado computador ou serviço através de inúmeras tentativas de descobrir a conta e senha de um usuário válido. A principal vulnerabilidade explorada por este ataque é humana e reside no fato de que os usuários tipicamente utilizam senhas fáceis de serem lembradas. Todo e qualquer serviço de rede que empregue autenticação de usuário é vulnerável a este tipo de ataque. Diversas ferramentas – “Lophtcrack”, “John the Ripper” etc - existem para realizar automaticamente as tentativas de autenticação. Estas ferramentas fazem uso de dicionários de palavras no idioma do usuário para agilizar a descoberta da senha – daí o nome do padrão de ataque. Se o usuário utilizar uma palavra existente no dicionário a ferramenta será capaz de identificar sua senha mais rapidamente do que em tentativas por força bruta (tentativa de todas as combinações de caracteres alfanuméricos até um determinado tamanho máximo de senha). No projeto MIT/DARPA um script chamando “NetGuess” realizava entre 10 e 100 tentativas de autenticação nos serviços ftp, pop e telnet utilizando um arquivo de dicionário e permutações simples envolvendo o nome do usuário.

Para que sistemas de detecção de intrusão possam detectar este tipo de ataque eles precisam possuir conhecimento prévio dos protocolos empregados por cada aplicação que possua autenticação remota. Basta então monitorar a quantidade de erros de autenticação em determinado intervalo de tempo para acusar uma intrusão deste tipo.

C.2 FTP-Write R-c-U

Padrão de ataque que explora falha de configuração em serviços FTP que fornecem acesso para usuários não autenticados (anônimos). Se o diretório raiz de um servidor FTP tiver como proprietário o usuário “ftp” ou seu proprietário estiver no mesmo grupo do usuário “ftp” é possível que um usuário anônimo tenha acesso para escrita de arquivos no servidor FTP e até mesmo obtenha acesso remoto ao servidor [42].

C.3 Guest R-c-U

Variante do padrão de ataque Dictionary descrito anteriormente, no ataque Guest é realizado a tentativa de acesso não autorizado utilizando apenas a conta de usuário “guest”. Esta conta de usuário existe em diversos sistemas operacionais modernos e usualmente é habilitada em instalação padrão destes sistemas.

C.4 Imap R-b-S

Imap é um ataque que explora uma falha de estouro de “buffer” (“buffer overflow”) [43] em serviço Imap de servidores RedHat Linux 4.2 e que permite ao atacante a execução de código arbitrário no servidor afetado. Uma ferramenta disponibilizada na Internet com instruções de uso e denominada “Impack 1.03 Attack Toolkit” [44] foi utilizada no projeto DARPA/MIT. A existência de ferramentas com esta tornam o ataque ainda mais perigoso uma vez que usuários sem conhecimentos técnicos sobre estouro de “buffer” podem seguir as instruções de uso e realizar com sucesso um ataque deste tipo.

C.5 NameD R-b-S

Ataque que explora uma falha de estouro de “buffer” [45] em determinadas versões de servidores DNS – Domain Name System – Bind. Uma consulta de informações de DNS reverso – obter informações sobre nome, dado um endereço IP - especialmente preparada pode tornar o serviço inoperante ou permitir ao atacante a execução de código no servidor.

Um servidor de DNS Bind aloca uma área de memória de no máximo 4096 bytes para uma consulta de DNS reverso. Este padrão de ataque constrói uma

consulta que extrapola este limite causando um estouro de “buffer”. Um sistema de detecção de intrusão de redes pode, para detectar este tipo de ataque, monitorar consultas deste tipo que ultrapassem 4096 bytes em sua área de dados.

C.6 Phf R-b-U

Phf [46] é um exemplo de programa CGI – “Common Gateway Interface”, mecanismo de interação entre aplicações em um servidor Web e programas do sistema operacional – disponibilizado com todo servidor Web Apache. Enviando parâmetros especialmente criados para comprometer este programa um atacante pode, por exemplo, copiar o arquivo “passwd” com os usuários e senhas de um sistema Unix vulnerável.

C.7 Sendmail R-b-S

O ataque Sendmail [47] explora uma falha de estouro de “buffer” na versão 8.8.3 do software de correio eletrônico SMTP que permite a execução de código com privilegio de administrador/”root”. Através do envio de uma mensagem de e-mail para o servidor alvo, especialmente construída para causar o estouro de “buffer”, o atacante consegue forçar o serviço sendmail a executar código de sua escolha.

C.8 XLock R-cs-Intercept (Keystrokes)

Neste ataque, o atacante irá enganar o usuário de uma sessão gráfica X a digitar sua senha em uma versão alterada (cavalo de Tróia) do programa Xlock – utilizado para bloquear uma sessão gráfica X após determinado período de inatividade. Uma versão modificada do programa Xlock foi desenvolvida pelo projeto MIT/DARPA especialmente para simular este ataque. Para este ataque ser bem sucedido o atacante tem que ter algum tipo de acesso a máquina do usuário para instalar a versão alterada do programa Xlock.

C.9 XSnoop R-c-Intercept (Keystrokes)

Neste ataque, o atacante monitora todas as teclas digitadas e enviadas por usuários para um servidor X desprotegido. Este registro de teclas enviadas pode conter informações confidenciais que serão úteis ao atacante em novos ataques.

Apêndice D
Padrões de ataque – Usuário para Super-usuario

7 padrões de ataques da classe “Usuário para Super-usuário” (“User to Root”) estão presentes na base de dados de treinamento e testes desenvolvida pelo MIT/DARPA e utilizada neste trabalho. São eles :

D.1 Eject U-b-S

Explora uma falha de estouro de “buffer” existente no utilitário “eject” disponibilizado no sistema operacional Solaris 2.5. Eject é um utilitário para manuseio de mídias removíveis. Um usuário com privilégios simples em um sistema vulnerável que explore esta falha passará a ter privilégios de administração e supervisão (“root”) [48].

D.2 Ffbconfig U-b-S

Bastante similar ao ataque Eject descrito anteriormente, este padrão de ataque explora um estouro de “buffer” no programa Ffbconfig utilizado no sistema Solaris 2.5 para configurar placas gráficas instaladas no sistema. Devido a não verificação de tamanho de parâmetros recebidos pelo programa é possível que um atacante sobrescreva áreas de memória interna do programa obtendo acesso não autorizado [49].

D.3 Fdformat U-b-S

Novamente uma falha de estouro de “buffer” é explorada para elevar o privilégio de usuário comum para usuário com poder de administração. O programa fdformat, vulnerável a este ataque, é utilizado para formatar disquetes

e cartões de memória PCMCIA. Código fonte que implementa este ataque foi disponibilizado em [49] e utilizado sem alterações no projeto MIT/DARPA.

D.4 LoadModule U-b-S

Ataque contra sistema operacional SunOS 4.1 que utilizem o sistema xnews de janelas. O programa xnews utiliza o módulo loadmodule para carregar alguns dispositivos em memória. Devido a um erro de programação no módulo loadmodule um atacante pode obter privilégios indevidos de administração.

D.5 Perl U-b-S

Explora falha em algumas versões da linguagem de scripts Perl. Um módulo denominado suidperl apresenta erro de programação que permite que qualquer usuário com conta ativa no sistema alvo obtenha privilégios de administração [50].

D.6 PS U-b-S

Ataque que explora uma falha no utilitário PS do sistema operacional Solaris 2.5 [51] . Se o usuário tiver permissão de escrita em diretórios temporários ele pode explorar esta falha para executar código com privilégios de administração no sistema alvo.

D.7 XTerm U-b-S

Explora uma falha que permite estouro de “buffer” na biblioteca Xaw do programa xTerm em versões de sistema operacional RedHat Linux. Permite ao atacante obter privilégios de administração supervisão a partir de uma conta de usuário comum.

Apêndice E
Código fonte desenvolvido

E.1 – Tratamento dos dados

// Gera tabelas temporárias para os campos service, protocol, flag e label com associação numérica única.

```
select identity(int,1,1) as id_service, service into services from kddcup group by service

select identity(int,1,1) as id_protocol, protocol_type into protocol from kddcup group by protocol_type

select identity(int,1,1) as id_flag, flag into flags from kddcup group by flag

select identity(int,1,1) as id_label, label into labels from kddcup group by label
```

// Gera cópia (kddcup2) da tabela original kddcup e atualiza todos os campos não numéricos para os respectivos campos numéricos presentes nas tabelas temporárias.

```
select * into kddcup2 from kddcup

update kddcup2
set kddcup2.protocol_type = p.id_protocol
from kddcup2, protocol p
where kddcup2.protocol_type=p.protocol_type

update kddcup2
set kddcup2.service=s.id_service
from kddcup2, service s
where kddcup2.service=s.service

update kddcup2
set kddcup2.flag=f.id_flag
from kddcup2, flags f
where kddcup2.flag=f.flag
```

```
update kddcup2
set kddcup2.label=l.id_label
from kddcup2, label l
where kddcup2.label=l.label
```

// Gera subconjuntos – ds1, ds2, ds3, ds4 e ds5 – para treinamento a partir da tabela original kddcup

```
// Gera DataSet1 - Todos os ataques e normal.

select top 10000 * into dataset1 from randomdata
order by newid()

insert dataset1 select * from randomdata
where label in ('phf.','','','','')

insert dataset1 select top 20 percent *
from randomdata where label in ('guess_passwd.')

// Gera DataSet2 - Ataques da categoria Probe e Normal

select top 10000 * into dataset2 from randomdata
where label in (select label from label where categoria = 'normal' or categoria = 'probe')
order by newid()

// Gera Dataset3 - Ataques da categoria DOS e Normal

select top 10000 * into dataset3 from randomdata
where label in (select label from label where categoria = 'normal' or categoria = 'dos')
order by newid()

// Gera DataSet4 - Ataques da categoria U2R e Normal

select top 10000 * into dataset4 from randomdata
where label in (select label from label where categoria = 'normal' or categoria = 'u2r')
order by newid()

// Gera DataSet5 - Ataques da categoria R2L e Normal

select top 10000 * into dataset5 from randomdata
where label in (select label from label where categoria = 'normal' or categoria = 'r2l')
order by newid()
```

// Etapa final de preparação realizada usando o Matlab. Consiste basicamente da carga de um subconjunto em uma matriz do Matlab e a normalização dos valores numéricos do subconjunto dentro do intervalo [-1; 1] através da função premnmx() do Matlab.

```
load('c:\Datasets\ds1.txt');

I = ds1(:,1:9);     % Caracteristicas Intrinsecas (I)
E = ds1(:,10:23);    % Características Especialista (E)
T = ds1(:,24:41);   % Características Temporais (T)
A = ds1(:,1:41);    % Todas as caracteristicas (All)
S = ds1(:,42);      % Saída Desejada

[In,mini,maxi]=premnmx(I');
[En,mine,maxe]=premnmx(E');
[Tn,mint,maxt]=premnmx(T');
[Sn,mins,maxs]=premnmx(S');
```

E.2 – Treinamento de rede neural

// Define parâmetros, cria o objeto Rede Neural MLP e treina a rede com dados carregados e normalizados anteriormente.

```
% define parametros da Rede Neural
neu_hidden1 = 15;
neu_hidden2 = 15;
neu_exit = 1;
num_epochs = 1500;

% Cria a rede
net=newff(minmax(In),[neu_hidden1 neu_hidden2 neu_exit],{'tansig','tansig', 'tansig'});
net = init(net);
net.trainParam.epochs=num_epochs;
net.trainParam.goal=0.01;
net.trainParam.show=25;

% Treina a Rede Neural com entradas P e saída desejada T
net=train(net,In,SS);
```

E.3 – Simulação e avaliação de resultados

// Carrega base de dados de teste (testdata.txt),normaliza dados de teste no intervalo [-1;1], simula a rede neural treinada anteriormente com esta base de dados de teste e avalia resultados calculando matriz de confusão. O código apresentado foi empregado para o cenário de identificação da classe – normal, probe, dos, r2l ou u2r - de cada registro de testdata.txt.

```
load 'c:\mestrado\dados\testdata.txt'

TI = testdata(:,1:41);
TIn = premnmx(TI');

TEMP = testdata(:,42);

TSS=zeros(max(size(TEMP)),5);

% Prepara matriz com resultado esperado para comparação com saída da rede.

for i=1:max(size(TEMP))
          if TEMP(i)== 2 | TEMP(i)==7 | TEMP(i)==13 | TEMP(i)==15 | TEMP(i)==16 |
          TEMP(i)==26,
                    TSS(i,:)=[-1 -1 -1 -1 1];
            elseif TEMP(i)== 6 | TEMP(i)== 10 | TEMP(i)== 11 | TEMP(i)== 24| TEMP(i)==28|
          TEMP(i)==30| TEMP(i)==34,
                    TSS(i,:)=[-1 -1 -1 1 -1];
            elseif TEMP(i)== 4 | TEMP(i)== 5 | TEMP(i)== 17 | TEMP(i)== 20 | TEMP(i)== 21 |
          TEMP(i)== 22| TEMP(i)==23| TEMP(i)==31| TEMP(i)==35| TEMP(i)==36
                    TSS(i,:)=[-1 -1 1 -1 -1];
            elseif TEMP(i)==27
                    TSS(i,:)=[1 -1 -1 -1 -1];
            else TSS(i,:)=[-1 1 -1 -1 -1];
           end
end

TT = sim(net,TIn);
TT = TT';

% Constroi Matriz de Confusão

CM = zeros(min(size(TSS))),zeros(min(size(TSS)));
```

```
for i=1:max(size(TT))
        if TSS(i,:)==[-1 -1 -1 -1 1]          % probe
                if sign(TT(i,:))==TSS(i,:)
                        CM(5,5)=CM(5,5)+1;
                elseif sign(TT(i,:))==[-1 -1 -1 1 -1]
                        CM(5,4)=CM(5,4)+1
                elseif sign(TT(i,:))==[-1 -1 1 -1 -1]
                        CM(5,3)=CM(5,3)+1;
                elseif sign(TT(i,:))==[-1 1 -1 -1 -1]
                        CM(5,2)=CM(5,2)+1;
                else sign(TT(i,:))==[1 -1 -1 -1 -1]
                        CM(5,1)=CM(5,1)+1;
                end

        elseif TSS(i,:)==[-1 -1 -1 1 -1]       % u2r
                if sign(TT(i,:))==TSS(i,:)
                        CM(4,4)=CM(4,4)+1;
                elseif sign(TT(i,:))==[-1 -1 -1 -1 1]
                        CM(4,5)=CM(4,5)+1
                elseif sign(TT(i,:))==[-1 -1 1 -1 -1]
                        CM(4,3)=CM(4,3)+1;
                elseif sign(TT(i,:))==[-1 1 -1 -1 -1]
                        CM(4,2)=CM(4,2)+1;
                else sign(TT(i,:))==[1 -1 -1 -1 -1]
                        CM(4,1)=CM(4,1)+1;
                end

        elseif TSS(i,:)==[-1 -1 1 -1 -1]       % dos
                if sign(TT(i,:))==TSS(i,:)
                        CM(3,3)=CM(3,3)+1;
                elseif sign(TT(i,:))==[-1 -1 -1 -1 1]
                        CM(3,5)=CM(3,5)+1
                elseif sign(TT(i,:))==[-1 -1 -1 1 -1]
                        CM(3,4)=CM(3,4)+1;
                elseif sign(TT(i,:))==[-1 1 -1 -1 -1]
                        CM(3,2)=CM(3,2)+1;
                else sign(TT(i,:))==[1 -1 -1 -1 -1]
                        CM(3,1)=CM(3,1)+1;
                end

        elseif TSS(i,:)==[-1 1 -1 -1 -1]       % r2l
                if sign(TT(i,:))==TSS(i,:)
                        CM(2,2)=CM(2,2)+1;
                elseif sign(TT(i,:))==[-1 -1 -1 -1 1]
                        CM(2,5)=CM(2,5)+1
```

```
                elseif sign(TT(i,:))==[-1 -1 -1 1 -1]
                        CM(2,4)=CM(2,4)+1;
                elseif sign(TT(i,:))==[-1 -1 1 -1 -1]
                        CM(2,3)=CM(2,3)+1;
                else sign(TT(i,:))==[1 -1 -1 -1 -1]
                        CM(2,1)=CM(2,1)+1;
                end

        elseif TSS(i,:)==[1 -1 -1 -1 -1]        % normal
                if sign(TT(i,:))==TSS(i,:)
                        CM(1,1)=CM(1,1)+1;
                elseif sign(TT(i,:))==[-1 -1 -1 -1 1]
                        CM(1,5)=CM(1,5)+1
                elseif sign(TT(i,:))==[-1 -1 -1 1 -1]
                        CM(1,4)=CM(1,4)+1;
                elseif sign(TT(I,:))==[-1 -1 1 -1 -1]
                        CM(1,3)=CM(1,3)+1;
                else sign(TT(i,:))==[-1 1 -1 -1 -1]
                        CM(1,2)=CM(1,2)+1;
                end
        end
end
```