Detecção de Backdoors e Canais Dissimulados

Carlos Henrique P. C. Chaves
LAC/INPE
Av. dos Astronautas, 1758
12227-010 São José dos Campos - SP
cae@lac.inpe.br

Antonio Montes
CenPRA/MCT
Rodovia Dom Pedro I, km 143,6
13082-120 Campinas - SP
antonio.montes@cenpra.gov.br

Resumo

Este artigo apresenta os conceitos de backdoors e canais dissimulados, alguns exemplos de ferramentas disponíveis publicamente e uma metodologia para detectá-los. O sistema de detecção de intrusão apresentado é um sistema híbrido, que utiliza mapas auto-organizáveis como técnica de detecção por anomalia e reconhecimento de padrões como técnica de detecção por abuso. O sistema tem como objetivo contribuir com toda comunidade de administração e segurança de redes, utilizando-se das vantagens das estratégias de detecção.

Palavras-chave: *segurança, backdoor, covert channel, detecção de intrusão*

1. Introdução

No mundo atual, a maioria das empresas e corporações possui uma rede local com acesso à Internet e a utiliza para expor seu produtos (através de páginas em servidores HTTP), para entrar em contato com os clientes (através de troca de correios eletrônicos) ou mesmo para pedir aos fornecedores renovação de estoque (através da comunicação entre seus computadores). Essa situação gera uma preocupação, por parte da empresa, com a segurança de seus dados e informações sigilosas. Para tentar resolver este problema, são adotadas tecnologias de segurança da informação, como *firewalls*, antivírus e sistemas de detecção de intrusão.

Existe uma tendência de crescente preocupação com a segurança de organizações em relação a ataques vindos da Internet que possam tornar indisponíveis os servidores da empresa ou que possibilitem o roubo de dados de caráter confidencial. Porém, a ameaça interna é real e iminente. Funcionários insatisfeitos ou que, por ventura, tenham sido demitidos podem representar um grande perigo à empresa. Não seria uma tarefa muito difícil para um destes funcionários instalar um *backdoor* ou um *rootkit* na máquina que ele utiliza (ou utilizava), que permita acessá-la com o intuito de prejudicar a organização, ou mesmo obter dados que possam render algum benefício com organizações concorrentes. *Backdoors* são programas executados em computadores com o objetivo de prover acesso aos mesmos sem que haja a necessidade de exploração de alguma vulnerabilidade. Eles podem ser detectados por sistemas antivírus, anti-*spyware* e de detecção de intrusão, uma vez que o tráfego destinado a eles normalmente contém padrões conhecidos.

Um outro aspecto importante consiste no fato de que os administradores de *firewall* normalmente definem regras para permitir o acesso, a partir da Internet, a apenas certas aplicações servidoras, como HTTP(S), SMTP, DNS, entre outros. As barreiras criadas por essas regras podem ser violadas com o auxílio de ferramentas que utilizam canais dissimulados (*covert channels*). Essas ferramentas tem como objetivo tentar transmitir informações de maneira que mecanismos de controle de acesso a redes e detecção de intrusão não consigam identificá-las.

Este artigo tem como objetivo apresentar um sistema de detecção de *backdoors* e canais dissimulados, que implementa técnicas de detecção de intrusão por anomalia e abuso para classificar um tráfego como pertencente a um certo *backdoor* ou ferramenta que implementa canais dissimulados. Além disso, o sistema visa superar os possíveis falso-negativos de um sistema de detecção por abuso, causados por eventuais alterações nos padrões das ferramentas de ataque, que são identificadas pelas regras de detecção. As ferramentas podem então ser detectadas por anomalia, uma vez que elas serão identificadas pelos seus comportamentos, e não por assinaturas em seus conteúdos. Este sistema também poderá ser utilizado para descobrir novos ataques, previamente não detectados por um sistema de detecção baseado somente em assinaturas.

O restante deste artigo está organizado da seguinte

forma: a seção 2 conceitua *backdoors*; a seção 3 define um canal dissimulado; a seção 4 conceitua sistemas de detecção de intrusão e apresenta técnicas de detecção; a seção 5 descreve o sistema de detecção de *backdoors* e canais dissimulados; a seção 6 mostra os resultados obtidos e, finalmente, a seção 7 apresenta as conclusões sobre o assunto abordado e os próximos trabalhos a serem desenvolvidos.

2. Backdoors

Os atacantes, após terem comprometido um sistema, normalmente utilizam um mecanismo para conseguir acesso a esse sem que uma vulnerabilidade em um software tenha que ser explorada. Este mecanismo é chamado de *backdoor*. Esse é freqüentemente instalado com a intenção de facilitar o retorno do atacante, bem como dificultar a sua detecção.

Um *backdoor* é normalmente instalado através da adição de um novo serviço ao sistema ou em substituição de um serviço legítimo por outro que atenda às necessidades do atacante.

3. Canais Dissimulados

Existem várias definições para um canal dissimulado (*covert channel*). Lampson conceitua canais dissimulados como sendo canais usados para troca de informações que não são normalmente utilizados para comunicação e que não são protegidos por mecanismos de controle de acessos, ou seja, uma comunicação ilícita através de um canal de troca de informações legítimo. Ele também chama os canais dissimulados de "*leakage paths*" [5]. Mudge o define como sendo o processo no qual um canal de comunicação é encapsulado por outro [7].

Uma vez que o protocolo HTTP é muito utilizado nas redes das corporações e normalmente não é filtrado por filtro de pacotes, este artigo irá considerar os canais dissimulados sobre o protocolo HTTP. É importante salientar que o simples fato de utilizar as portas associadas ao HTTP para trafegar outras informações não caracteriza o uso de um canal dissimulado. Para isso, é necessário que outras informações estejam encapsuladas no protocolo HTTP.

4. Sistemas de Detecção de Intrusão

Os sistemas de detecção de intrusão podem ser definidos como mecanismos que permitem detectar a exploração de falhas de segurança, ou seja, detectar ataques. Segundo Sundaram [11], as técnicas de detecção de intrusão são divididas em duas categorias principais: detecção de intrusão por anomalia e detecção de intrusão por abuso.

As técnicas de detecção por anomalia assumem que todas as atividades intrusivas são necessariamente anômalas. Isto significa que se um perfil de uma atividade normal puder ser criado, é possível (em teoria) que todos os eventos que variam de forma estatisticamente significativa desse perfil sejam classificados como tentativas de intrusão. Entretanto, se for considerado que o conjunto de atividades intrusivas apenas intercepte o conjunto de atividades anômalas ao invés de ser exatamente o mesmo, então existem algumas possibilidades: atividades anômalas que não são intrusivas e são classificadas como intrusivas (falso-positivos); atividades intrusivas que não são anômalas e não são classificadas como intrusivas (falso-negativos) [11].

As técnicas de detecção por abuso se baseiam na afirmativa de que existem meios de representar um ataque utilizando padrões ou assinaturas, de forma que variações do mesmo ataque possam ser detectadas. As questões primordiais em sistemas de detecção por abuso estão em como escrever uma assinatura que cerque todas as variações de um determinado ataque, e também em como escrever assinaturas que não casem com atividades não intrusivas. Uma limitação desse método consiste no fato de que ele busca por vulnerabilidades conhecidas, sendo que novas vulnerabilidades não são reconhecidas como intrusões [11].

5. Sistema de Detecção de *Backdoors* e Canais Dissimulados

A idéia inicial do sistema de detecção de *backdoors* e canais dissimulados surgiu com a criação de uma metodologia de detecção proposta por Chaves e Montes [2], baseada na análise das sessões TCP/IP do tráfego de uma rede.

A metodologia é dividida em três fases: a reconstrução das sessões TCP/IP, a análise e classificação, e a geração do resultado.

A primeira fase propõe a utilização do Sistema de Reconstrução de Sessões TCP/IP (Recon) [3] para efetuar a reconstrução das sessões TCP/IP, e, além disso, a extensão do Recon, de modo que ele passe a analisar uma quantidade configurável de dados, para tornar possível a busca por assinaturas no conteúdo dos pacotes. Na segunda fase, uma vez reconstruída, a sessão é analisada em busca de características que a classifiquem como pertencente a um *backdoor* ou canal dissimulado. Primeiramente é feita uma análise de comportamento do protocolo utilizado na sessão. Depois, procura-se no conteúdo do pacote por assinaturas conhecidas dos *backdoors* e das ferramentas que implementam canais dissimulados, de forma a identificá-los. A terceira fase da metodologia gera um relatório contendo o resultado da análise e classificação feita na segunda fase e as informações das sessões TCP/IP [2].

Essa metodologia leva à criação de um sistema de detecção de intrusão híbrido (que realiza tanto a detecção por anomalia quanto por abuso), sendo feita primeiramente a detecção por anomalia e posteriormente por abuso. Neste artigo, serão discutidas apenas as sessões que utilizam o protocolo TCP na camada de transporte (porém as idéias podem ser facilmente implementadas para outros protocolos).

Uma vez que a base do sistema está formada, o próximo passo caracteriza-se no desenvolvimento do mecanismo de detecção de *backdoors* e canais dissimulados. O princípio básico para detecção de um *backdoor* está em encontrar características que indiquem a atividade de interesse. Entre os candidatos para essas características, estão: as assinaturas nos dados, o tamanho e a taxa de transmissão dos pacotes e o intervalo de tempo entre os pacotes de uma dada sessão [13]. Em relação a detecção de canais dissimulados, mais especificamente os que utilizam o protocolo HTTP, alguns autores concordam que a definição de limiares para certas características das sessões HTTP são suficientes para detecção de canais dissimulados. Além disso, eles concordam que essa idéia é aplicável a outros protocolos da pilha TCP/IP [1, 7, 8].

As idéias pesquisadas, em adição ao objetivo de desenvolver um sistema de detecção de intrusão por anomalia, levaram à busca por trabalhos com objetivos semelhantes ao desse. Conforme dito na seção 4, a detecção de intrusão por anomalia se baseia na definição de um perfil normal e na comparação das informações dos eventos com o normal. Sendo assim, a técnica difere entre os sistemas na maneira de representar o perfil e em como calcular a diferença entre o normal e os eventos observados. Para isso, podem ser utilizadas técnicas probabilísticas [12], técnicas baseadas em análise de agrupamentos (*clustering*) [9], técnicas de *Data Mining* [6] ou redes neurais artificiais [10].

Para representar uma sessão, nove características foram selecionadas, sendo estas:

1. Tamanho médio dos pacotes recebidos pelo cliente;
2. Tamanho médio dos pacotes recebidos pelo servidor;
3. Número de pacotes recebidos pelo cliente;
4. Número de pacotes recebidos pelo servidor;
5. Porcentagem de pacotes pequenos;
6. Direção do tráfego;
7. Total de dados recebidos pelo cliente;
8. Total de dados recebidos pelo servidor;
9. Duração da sessão.

Para cada sessão, essas características são extraídas de forma a gerar perfis normais do tráfego. Uma vez que cada protocolo pode gerar vários perfis normais com certa diferença entre eles, o mecanismo de detecção de *backdoors* e canais dissimulados irá utilizar, para representar esses perfis e detectar sessões anômalas, mapas auto-organizáveis (SOM) [4].

Uma vez que as características utilizadas estão em ordem de grandeza e unidades diferentes, é feita uma normalização dos dados. Este processo é dividido em duas etapas, sendo que a primeira consiste no cálculo da média (μ) e do desvio padrão (σ) para cada uma das características. Na segunda etapa, cada vetor de nove posições $< c1, c2, c3, c4, c5, c6, c7, c8, c9 >$, contendo as características das sessões, é normalizado para $< n1, n2, n3, n4, n5, n6, n7, n8, n9 >$, utilizando-se a seguinte equação:

$$n_i = \frac{c_i - \mu_i}{\sigma_i} \quad (1)$$

O SOM utilizado no sistema é bidimensional, e cada neurônio é representado como um vetor de nove dimensões, formados pelas características das sessões de forma normalizada. O mapa precisa ser primeiramente treinado, como qualquer rede neural artificial. Para isso, um tráfego legítimo (sem nenhum pacote ou sessão maliciosa) deve ser utilizado.

A fase de aprendizagem (treinamento) começa com a inicialização do SOM com valores aleatórios entre 0 e 1. Depois, para cada vetor (instância) das sessões, um algoritmo competitivo é utilizado para encontrar o nó vencedor na rede (aquele que possui menor distância euclidiana em relação a instância avaliada). Ao ser encontrado, o nó vencedor e seus vizinhos dentro de um certo raio ou vizinhança atualizam seus pesos (através de uma função de aprendizagem) para representar a classe do padrão de entrada. Ao final dessa fase, o SOM está treinado e pronto para ser utilizado para o reconhecimento de padrões (fase de operação), ou seja, para detecção de sessões anômalas.

A fase de operação do SOM acontece da seguinte forma: para cada sessão extraída pelo Recon, é criado um vetor (instância) contendo as nove características utilizadas. Esse vetor é normalizado utilizando-se a equação 1, e depois o nó vencedor (com menor distância euclidiana) é encontrado. A sessão será classificada como normal se ela estiver suficientemente perto do nó vencedor, e anômala caso a distância para o nó vencedor seja maior que um limiar pré-definido. A definição deste limiar é feita por tentativa e erro, de modo que o número de falso-positivos e falso-negativos sejam mínimos. Ao final dessa fase, todas as sessões TCP foram classificadas como normais ou anômalas.

Após a detecção por anomalia, é feita a detecção por abuso, onde o conteúdo das sessões classificadas como anômalas são analisados em busca de assinatu-

ras de ataques conhecidos. O mecanismo de detecção utiliza regras baseadas nas utilizadas pelo sistema de detecção de intrusão *Snort*[1]. Essa estratégia visa a identificação do *backdoor* ou ferramenta de canal dissimulado utilizada no ataque.

Finalmente, o sistema de detecção de *backdoors* e canais dissimulados gera um relatório detalhado contendo todos os dados da sessões TCP/IP, incluindo a sua classificação (normal ou anômala) e os alertas gerados pelo mecanismo de detecção por abuso. Este relatório deverá ser examinado por um analista para uma estudo mais detalhado sobre as sessões maliciosas.

6. Resultados

O sistema de detecção de *backdoors* e canais dissimulados foi testado utilizando tráfego do protocolo HTTP da rede do Laboratório Associado de Computação a Matemática Aplicada (LAC) do INPE. Para sua coleta, um sensor foi posicionado no mesmo segmento da interface interna do *firewall*, e foi capturado o tráfego de um mês (julho de 2005). O tráfego referente a primeira semana de julho, entre os dias 03 e 09, foram utilizados para treinar o sistema de detecção por anomalia. O perfil do tráfego pode variar normalmente, de acordo com o dia da semana e o período de cada dia, ou seja, o perfil do tráfego da madrugada de domingo não deve ser parecido com o perfil do horário comercial de um dia útil. Sendo assim, foram gerados quatro arquivos por dia, de acordo com os horários de trabalho no LAC (0:00-7:59, 8:00-12:59, 13:00-17:59 e 18:00-23:59). Na fase de teste, o perfil referente ao dia e período correto deve ser utilizado, ou seja, se o tráfego analisado for de segunda-feira, entre 10:00 e 11:00 horas, então o perfil utilizado será de segunda-feira, de 8:00 às 12:59.

Para testar o sistema, foi utilizado como estação de análise um computador com processador Intel Pentium 4 de 2.66GHz, 512Kb de memória cache, 512Mb de memória RAM e o sistema operacional Slackware Linux 10.1. Um fator importante a ser colocado é o fato de toda a reconstrução e análise das sessões TCP/IP ser feita em memória. Assim, o desempenho do sistema é dependente do total de memória RAM livre no momento da sua execução. Se o arquivo contendo as sessões TCP/IP ultrapassar a quantidade de memória disponível, fatalmente a área de *swap* será utilizada, e o desempenho do sistema será afetado. Sendo assim, existe uma limitação em relação ao tamanho do arquivo a ser analisado.

O primeiro teste realizado no sistema foi em busca de falso-positivos, utilizando como entrada os arquivos usados para gerar os perfis do tráfego. Além de informar o número de falso-positivos, este teste valida o treinamento da rede neural. O sistema apresentou uma taxa média de 0,17% de sessões anômalas, ou seja, em um tráfego contendo apenas sessões lícitas, 0,17% delas são classificadas como anômalas pelo sistema de detecção, resultando em uma taxa de 0,17% de falso-positivos.

O próximo passo foi testar o mecanismo de detecção utilizando o tráfego entre os dias 10 e 30 do mês de julho, que compreende ao restante do mês analisado. Durante esses 21 dias, foram analisadas 761437 sessões HTTP, com uma média de 1,05% de sessões anômalas. O total de sessões analisadas por dia variou entre 3288 e 75713 (Figura 1), e a taxa de sessões anômalas variou entre 0,41% e 6,82% (Figura 2).

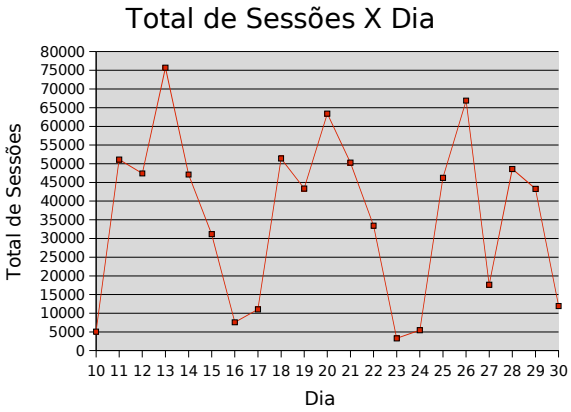

Figura 1. Total de sessões HTTP por dia.

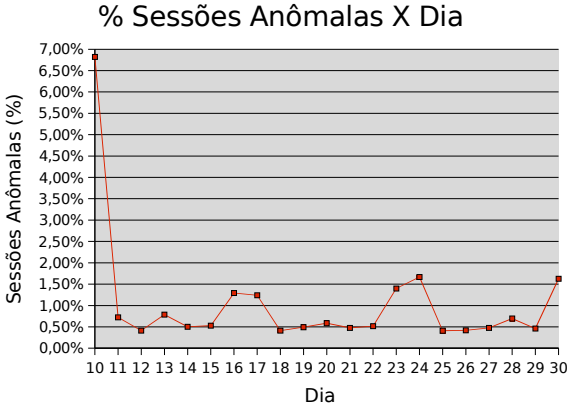

Figura 2. Percentagem de sessões anômalas por dia.

Nos testes realizados entre os dias 10 e 30 do mês de julho, não foram detectados acessos a *backdoors* ou a utilização de ferramentas que implementam canais dissimulados. Algumas sessões ilícitas (dependendo da política de segurança da empresa) referentes a utilização de MSN sobre HTTP (chat), acessos as

[1]http://www.snort.org

sítios pornográficos e *download* de softwares piratas foram detectadas. Sendo assim, para testar a detecção de *backdoors* e canais dissimulados, foram utilizadas duas máquinas: uma atuando como cliente, localizada na rede interna do LAC, e outra atuando como servidora, localizada fora da rede do LAC. O tráfego entre as máquinas foi capturado durante a execução de algumas ferramentas, e analisado pelo sistema de detecção de *backdoors* e canais dissimulados.

O *netcat*[2] foi utilizado para criar um *backdoor* no servidor, na porta 80/tcp. O cliente, na máquina da rede interna, executou vários comandos no servidor. O sistema reconstruiu uma sessão do *netcat*, sendo essa classificada como anômala. Como não foi definida uma assinatura que identifica o *netcat*, a ferramenta não foi identificada, porém um alerta foi disparado informando que o comando *id* retornou *root*, ou seja, o atacante tinha privilégio de super-usuário. Uma vez que comando *id* foi executado pelo cliente, e o *backdoor* executado com privilégio de súper-usuário, o resultado do comando foi *uid=0(root)*. Essa cadeia casou com a cadeia de caracteres procurada por uma regra de detecção: "*uid=0(root)*".

O *rwwwshell*[3] foi utilizado para simular um ataque de *shell* reverso, onde o atacante, no servidor externo, executa comandos na máquina localizada na rede interna. O sistema reconstruiu oito sessões, sendo apenas uma classificada como anômala. A assinatura que identifica a ferramenta foi encontrada na sessão anômala (*POST /cgi-bin/orderform*). Sete sessões referentes a comandos executados pelo atacante foram classificadas como normais, resultando em uma taxa de 87,5% de falso-negativos para o teste com essa ferramenta. Porém, este valor alto não é um problema, visto que todas as sessões, tanto anômalas quanto normais, pertencem ao mesmo par de IPs. Assim, todas deverão ser posteriormente examinadas pelo analista.

O último *backdoor* testado foi o *cctt*[4], sendo realizado o mesmo tipo de teste do *netcat*. O sistema reconstruiu uma sessão do *cctt*, sendo essa classificada como anômala. Neste modo de operação, o *cctt* envia diretamente os comandos para o servidor, não existindo assinaturas específicas. Como no caso no *netcat*, um alerta foi disparado informando que o comando *id* retornou *root*, ou seja, o atacante tinha privilégio de super-usuário.

As ferramentas *httptunnel*[5] e *firepass*[6] foram utilizadas para encapsular o tráfego de SSH sobre o HTTP, simulando um cliente na rede interna utilizando um canal dissimulado para enviar dados para fora da organização (espionagem industrial). O sistema reconstruiu duas sessões do *httptunnel*, e ambas foram classificadas como anômalas. Além disso, uma assinatura que identifica a ferramenta foi encontrada na primeira sessão (*POST /index.html?crap=*). A respeito do *firepass*, três sessões foram reconstruídas, sendo todas classificadas como anômalas. A assinatura que identifica a ferramenta foi encontrada em todas as sessões anômalas (*POST /cgi-bin/fpserver.cgi*).

A ferramenta *wsh*[7] foi utilizada para executar comandos no servidor *Web* fora da rede do LAC. Além disso, o *wsh* foi utilizado para transferir um arquivo do cliente para o servidor, através do canal dissimulado, simulando o ato de um atacante enviar um programa à máquina invadida para, por exemplo, elevar seu privilégio de usuário comum (*apache*) para super-usuário (*root*). O sistema reconstruiu sete sessões, sendo duas classificadas como anômalas (sessões referentes a transferência de dois arquivos para o servidor). A assinatura que identifica a ferramenta foi encontrada em todas as sessões anômalas (*POST /cgi-bin/wsh-s*). As sessões referentes ao envio de comandos para o servidor e o recebimento das respostas foram classificadas como normais, indicando uma taxa de 71% de falso-negativos para o teste com essa ferramenta. Porém, como no caso do *rwwwshell*, este valor alto não é um problema, pois todas as sessões pertencentes as duas máquinas deverão ser examinadas pelo analista.

7. Conclusões e Trabalhos Futuros

Os *backdoors* e canais dissimulados são problemas enfrentados por qualquer administrador de redes. O sistema desenvolvido para sua detecção pretende dar uma contribuição para toda comunidade de administração e segurança de redes.

A utilização de mapas auto-organizáveis (SOMs) como mecanismo de detecção por anomalia se mostrou eficiente para detecção de *backdoors* e canais dissimulados. A utilização deles em sistemas de detecção de intrusão não é inédita, porém a utilização em conjunto com um mecanismo de detecção por abuso se torna um diferencial entre os sistemas existentes. As assinaturas dos *backdoors* e canais dissimulados podem ser encontradas e assim, regras de detecção podem ser criadas. Porém, como é comum em todo sistema de detecção por abuso, as ferramentas podem ser alteradas para não serem mais identificadas pelas regras. Neste caso, as ferramentas podem ser identificadas pelo seu comportamento, e assim uma nova regra de detecção pode ser gerada de acordo com a nova assinatura.

É importante salientar que a criação de perfis para um determinado tráfego deve ser uma tarefa dinâmica. A característica do tráfego de uma rede pode mudar

[2] http://netcat.sourceforge.net/
[3] http://www.thc.org/releases/rwwwshell-2.0.pl.gz
[4] http://www.gray-world.net/pr_cctt.shtml
[5] http://www.nocrew.org/software/httptunnel.html
[6] http://www.gray-world.net/pr_firepass.shtml
[7] http://www.gray-world.net/pr_wsh.shtml

normalmente, sem que ocorram eventos maliciosos. Assim, os perfis devem ser ajustados a essa nova situação, a fim de reduzir o número de falso positivos.

O sistema considera apenas as sessões que utilizam o protocolo TCP na camada de transporte. Porém, ele foi desenvolvido de forma que a expansão para a análise de outros protocolos, como UDP e ICMP, pode ser feita com facilidade. Além disso, várias características observadas nas sessões TCP podem ser empregadas para a definição dos perfis normais de outros protocolos.

A técnica de detecção por abuso se baseia na busca de assinatura no conteúdo das sessões, e o mecanismo de detecção utiliza regras baseadas nas utilizadas pelo sistema de detecção de intrusão *Snort*. Este mecanismo foi desenvolvido, de maneira simples, para buscar por cadeias de caracteres no conteúdo das sessões. Para o próximo trabalho, é proposta a utilização de expressões regulares e busca por assinaturas em hexadecimal, como é feito no *Snort*.

O sistema apresentou um resultado satisfatório, pois pelo menos uma sessão referente a cada uma das ferramentas testadas foi detectada. Em algums casos houveram taxas elevadas de falso-negativos, onde apenas uma de várias sessões foram detectadas como anômalas. Porém, isto não representa um problema, pois o resultado do sistema sempre deve ser avaliado por um analista. Assim, ao analisar a sessão que foi detectada como anômala, o analista deverá também analisar as outras sessões entre um mesmo par de IPs. Como continuação, é proposto o teste do sistema com tráfego gerado por outros *backdoors* e ferramentas que implementam canais dissimulados.

A última consideração sobre os trabalhos futuros consiste na modificação do sistema para trabalhar em tempo real, armazenando o tráfego de rede coletado e gerando alertas a medida que as sessões anômalas são identificadas.

Referências

[1] Castro, S. "Covert Channel and Tunneling over the HTTP protocol Detection: GW implementation theoretical design". Gray World.net Team, 2003. `http://www.gray-world.net/projects/papers/html/cctde.html`.

[2] Chaves, C. H. P. C. & Montes, A. "Backdoors e Canais Dissimulados: uma metodologia para detecção". In *Anais do VI Simpósio sobre Segurança em Informática (SSI'2004)*. São José dos Campos, SP. 2004.

[3] Chaves, M. H. P. C. *Análise de Estado de Tráfego de Redes TCP/IP para Aplicação em Detecção de Intrusão*. Dissertação de mestrado, Instituto Nacional de Pesquisas Espaciais, Laboratório Associado de Computação e Matemática Aplicada LAC, 2002.

[4] Kohonen, T. *Self-Organizing and Associative Memory*. Springer-Verlag, 3 edn., 1989.

[5] Lampson, B. W. "A note on the confinement problem". *Communications of the ACM*, 1973, 16(10), 613–615.

[6] Lee, W. & Stolfo, S. "Data mining approaches for intrusion detection". In *Proceedings of the 7th USENIX Security Symposium*. 1998. `http://citeseer.ist.psu.edu/article/lee98data.html`.

[7] Mudge. "Insider threat". *;login: The Usenix Magazine*, 2003, 28(06).

[8] Pack, D. J.; Streilein, W.; Webster, S. & Cunningham, R. "Detecting HTTP Tunneling Activities". In *In Proceedings of the 2002 IEEE Workshop on Information Assurance*. 2002. `http://www.ll.mit.edu/IST/pubs/Pack-IEEE2002.pdf`.

[9] Portnoy, L.; Eskin, E. & Stolfo, S. "Intrusion detection with unlabeled data using clustering". In *ACM Workshop on Data Mining Applied to Security (DMSA 2001)*. 2001. `http://citeseer.ist.psu.edu/article/portnoy01intrusion.html`.

[10] Ramadas, M.; Ostermann, S. & Tjaden, B. C. "Detecting anomalous network traffic with self-organizing maps". In *RAID*. 2003. `http://www.cs.fit.edu/~pkc/id/related/ramadas03raid.ps.gz`.

[11] Sundaram, A. "An introduction to intrusion detection", 2000. `http://coast.cs.purdue.edu/pub/doc/intrusion_detection/Intrusion-Detection-Intro.ps.Z`.

[12] Ye, N.; Li, X.; Chen, Q.; Emran, S. M. & Xu, M. "Probabilistic techniques for intrusion detection based on computer audit data". *IEEE Transactions on Systems, Man, and Cybernetics*, 2001, 31(4).

[13] Zhang, Y. & Paxon, V. "Detecting Backdoors". In *Proceedings of the 9th Usenix Security Symposium*. 2000. `http://www-cse.ucsd.edu/~savage/cse291/papers/Zhang00-1.pdf`.