Detecção de Tentativas de Intrusão em Sistemas por Análise de Tráfego de Rede

Emiliano F. Castejon
Instituto Nacional de Pesquisas Espaciais
Laboratório Associado de Computação
castejon@lac.inpe.br

Antonio Montes
Instituto Nacional de Pesquisas Espaciais
Laboratório Associado de Computação
montes@lac.inpe.br

Resumo

Neste artigo é apresentado um modelo de um Sistema de Detecção de Tentativas de Intrusão em um sistema computacional que, partindo da captura de dados provenientes do tráfego de rede é capaz de gerar uma base de dados com o perfil do ambiente monitorado e detectar eventos anômalos pela posterior utilização de técnicas de data-mining associadas à algoritmos de aprendizado induzido.

1. Introdução

Um dos setores da área de informática que atualmente tem passado por grandes mudanças é o setor de comunicação de dados quando associado a redes de computadores e disponibilidade de produtos e serviços *on-line*.

A realidade que existe hoje é a de que na maioria das vezes, associado ao lado positivo destas mudanças, está associado um lado negativo constituído por falhas de implementação nestes sistemas. Uma das conseqüências diretas é o surgimento de brechas de segurança passíveis de serem exploradas, levando ao comprometimento da estabilidade e confiabilidade de sistemas. Em [ITL0599] são discutidas várias técnicas e tipos de ataques e tentativas de intrusão que exploram estas brechas de segurança.

Para tentar suprir esta falha e prevenir o comprometimento de sistemas existe a área de Segurança em Sistemas de Informação provendo metodologias e tecnologias dispostas em vários níveis, ou camadas, que são compostas pela criação de políticas específicas de segurança, implantação e configuração de sistemas, dentre outros.

Agindo em uma camada intermediária está a implantação de Sistemas de Detecção de Intrusão (SDI) que, monitorando as atividades executadas em um determinado sistema podem ser configurados para agir de modo preventivo ou de modo ativo. Geralmente no primeiro caso, se são detectados eventos anômalos que possivelmente podem levar a situações de características intrusivas, apenas alertas são emitidos. Já no segundo caso, não só alertas são emitidos como ações podem ser tomadas diretamente sobre o sistema. Tais ações pode ir de controle/limitação até bloqueio ao acesso de recursos . Em [ITL1199] é feita uma breve descrição sobre alguns dos principais modelos de Sistemas de Detecção utilizados.

1.1 Falsos Positivos

Graças à evolução da comunicação digital é possível transmitir maiores volumes de dados em períodos de tempo cada vez menores. Tais sistemas de transmissão de banda larga estão se tornando padrão tanto para a conexão comercial quanto doméstica. Esta característica torna bem mais difícil o trabalho de monitorar e garantir a segurança de sistemas computacionais, principalmente quando a tarefa exige supervisão humana. O trabalho de sistemas de detecção também é dificultado, dada a crescente quantidade de dados que por eles deve ser filtrada e analisada.

Maior volume de dados implica diretamente em maior ruído, ou seja, dados que não contribuem ou não são relevantes para a detecção de tentativas de intrusão mas que, de alguma forma, podem levar o SDI a falsos resultados positivos. Um SDI bem estruturado tem de alguma forma filtrar o ruído para que o resultado final retornado seja o mais correto possível, ou seja, a ocorrência de alarmes falsos seja mínima.

1.2 *Data-Mining* e Detecção de Intrusão

Data-Mining é uma técnica largamente utilizada atualmente para a análise e extração de características de grandes volumes de dados. Sua aplicação em bases de dados já é feita a bastante tempo e a partir de 1995 começou a ser utilizada em SDIs como é mostrado em [Wenke99] onde é demonstrada a eficácia de uma técnica derivada, denominada KDD (*Knowledge Discovery in Databases*) [Fay96] para o tratamento de grandes volumes de dados.

Outro benefício direto da utilização de *data-mining* é a

possibilidade de detecção de eventos esparsos e de longa duração, ou seja, eventos que podem envolver quantidades diminutas de dados e que estão aleatoriamente dispostos por um longo período de tempo no fluxo de dados capturados da rede. Esta é uma das principais características de varreduras ou mapeamentos de computadores e serviços disponíveis em uma rede, fase inicial de possíveis tentativas de intrusão.

2. Aquisição e Armazenamento de Dados

A primeira tarefa de um SDI é a obtenção de dados brutos para a análise. Para o modelo descrito nesse artigo os dados brutos são compostos pelo conteúdo de pacotes IP capturados do tráfego de rede.

A captura pode ser feita de várias formas. Uma das possíveis formas é a utilização de bibliotecas de captura, como a LibPcap[1] que disponibiliza todo o conteúdo do pacote em estruturas de memória prontas para o processamento. Outra forma alternativa para a aquisição de dados é a utilização de módulos sensores de sistemas de monitoramento de redes, como é o caso do sistema SHADOW[2]. Neste último caso todo o conteúdo dos pacotes é gravado em disco e organizado em arquivos de acordo com o período de tempo em que foi feita a captura, deste modo cabe ao SDI efetuar a varredura em tais arquivos.

2.1 O primeiro filtro de dados – Seleção de Conteúdo de Pacotes

O conteúdo do pacote IP pode ser dividido em campos de cabeçalho e dados encapsulados (onde estão os demais cabeçalhos de protocolos de transporte). Nem todo este conteúdo é necessário ou relevante para o SDI.

Dependendo do tipo de análise que será efetuada posteriormente a maioria dos campos de cabeçalho e até mesmo todos os dados encapsulados podem ser ignorados, como mostrado na figura 2.1.1. Desta forma somente alguns campos de cabeçalho são realmente importantes para o modelo descrito neste artigo. São eles:

- Cabeçalho IP: *Time-Stamp, Flags IP, IP Length, IP Fragment Offset, Transport Protocol*.
- Cabeçalho TCP: *Flags TCP, source port, destination port, destination address, source address*.
- Cabeçalho UDP: *Source port, source address, destination port, destination address*.
- Cabeçalho ICMP: *ICMP Type*.

[1] LibPCap - LBNL's Network Research Group http://www-nrg.ee.lbl.gov

[2] NSWC Shadow - http://www.nswc.navy.mil/ISSEC/CID/

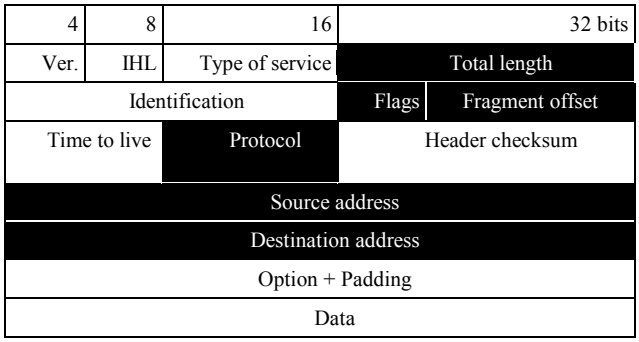

Figura 2.1.1 – Seleção de dados de cabeçalho IP

2.2 Armazenamento em base SQL

Mesmo com a seleção feita pela etapa anterior a quantidade de dados resultante ainda é extensa e desta forma deve ser armazenada em disco para posterior análise.

Visando um melhor desempenho tanto para armazenar quanto para recuperar as informações do disco é aconselhável a utilização de um sistema de banco de dados que se encarregue de toda a tarefa de otimização e organização destes dados em disco. Desta forma, para o modelo aqui citado, optou-se pela utilização de uma base de dados MySQL[3] que oferece alta portabilidade e possui bom desempenho para extensas quantidades de dados.

3. Extração de características do tráfego

Quando citados na área de detecção de intrusão por anomalia em tráfego de rede, padrões são considerados como uma forma de associar a ocorrência de eventos neste tráfego e, a partir disto, ser possível extrair alguma informação compreensível e útil de como pode-se caracterizar este tráfego com relação a possuir, ou não, características intrusivas.

A forma de se combinar dados para que um padrão seja montado varia de acordo com o tipo de característica que se deseja extrair dos dados sob análise. Em [Wenke98] é mostrada uma forma de montagem onde o objetivo é monitorar o andamento de conexões levando em consideração duração e quantidade de dados enviados.

Para o modelo citado neste artigo os padrões são montados de acordo com a idéia de "possível relação entre pacotes IP" estudando a forma de como a rede se comporta na ocorrência de um determinado tipo de pacote. Estão aqui inclusas todas as formas possíveis de reação (aceitações de conexões, respostas específicas de protocolos, rejeições de conexões, etc) que *hosts,* pertencentes a rede local monitorada, podem ter no evento de terem recebido (ou enviado) pacotes.

[3] MySQL – http://www.mysql.org

3.1 Geração de Padrões

A tarefa de combinar pacotes IP no tráfego de rede para gerar padrões é simples, entretanto alguns limites devem ser impostos. Em um conjunto de dados no formato LipCap/TCPDump, referente à um período de uma hora, por exemplo, a quantidade de pacotes é muito grande e portanto, se fosse feita a combinação destes pacotes dois a dois a geração de padrões nunca seria concluída em tempo hábil.

Em [Wenkee98] é mostrado um modelo onde os dados brutos são divididos em pequenos blocos de tamanho fixo. Assim, a combinação entre cada elemento é limitada a apenas integrantes de cada bloco. Esta forma de combinação resolve o problema citado anteriormente, entretanto, quando aplicada em tráfego de rede, não é eficiente pois a geração de alguns padrões importantes pode ser comprometida a medida que os pacotes IPs componentes destes padrões estejam em blocos de dados distintos e desta forma nunca serão combinados.

Deste modo, ao lidar com dados capturados de rede, é importante que se mantenha uma idéia de continuidade temporal, dada a própria característica da ocorrência seqüencial dos pacotes IP, cada um com seu próprio *time-stamp*.

Agregando esta idéia à forma de agrupamento de pacotes para a geração de padrões a combinação de dois pacotes pode ser limitada à uma janela de tempo deslizante que, varrendo todo o conjunto de dados de rede pacote a pacote, agrupa um conjunto de pacotes de modo que a diferença entre o *time-stamp* do ultimo e do primeiro pacote de mesma janela nunca seja maior que o um limite de tempo fixado como mostrado na figura 3.1.1.

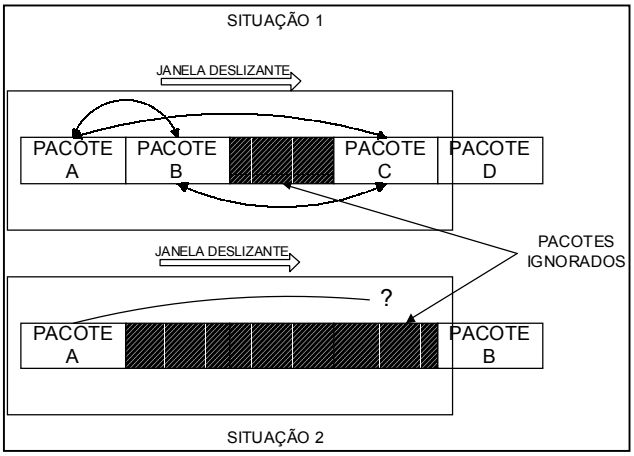

Figura 3.1.1 – Janela deslizante

Nesta situação o limite de tempo imposto pela janela funciona como um regulador de sensibilidade do sistema. Este limite deve ser fixado de acordo com os tipos de eventos que se deseja detectar. Pela análise manual do próprio tráfego de rede e também levando em consideração o protocolo padrão encontrado em RFCs[4] correspondentes a IP, TCP, UDP e ICMP é possível englobar todas as formas de reação citadas no item 3 com um limite de um segundo

Pode acontecer, dependendo da posição da janela nos dados do tráfego, que não seja possível combinar um pacote IP com algum outro. Em situações como esta apenas um pacote entra como parte integrante do padrão e um *flag* é configurado explicitando que não existe associação, como mostrado na situação 2 da figura 3.1.1. As demais configurações para este *flag* indicam as seguintes situações:

- Relacionamento completo: Os dois pacotes IP envolvidos fazem parte de um mesmo fluxo de dados bidirecional entre dois *hosts* específicos.
- Semi-relacionamento: Os dois pacotes envolvidos não fazem parte de um mesmo fluxo de dados entre dois *hosts* especificamente, mas estão relacionados à pelo menos um deles.

Outros dados agregados ao conteúdo de um padrão são (também mostrados na figura 3.1.1):

- *Time-stamp* do padrão: Dia da semana e período de hora do dia, de acordo com o valor do *time-stamp* do primeiro pacote IP agregado ao padrão.
- Número médio ponderado de ocorrências do padrão em determinado período do dia em determinado dia da semana.
- Número médio ponderado de ocorrências por janela em determinado período do dia em determinado dia da semana.

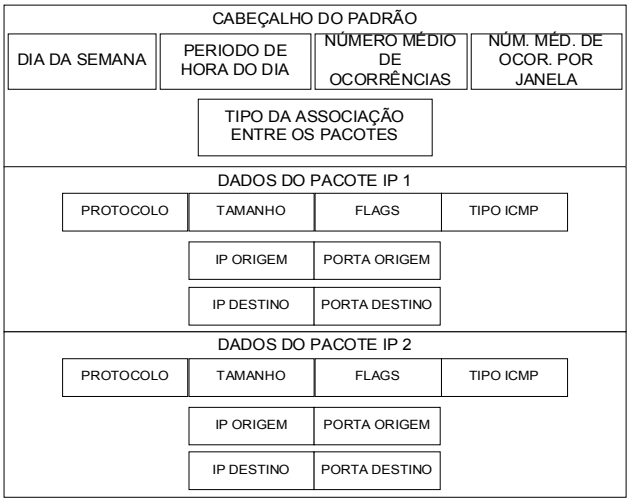

Figura 3.1.2 – Formato do Padrão

[4] RFC Search - http://www.faqs.org/rfcs/rfcsearch.html

3.2 O segundo filtro de dados – Seleção de *hosts* sob monitoramento

Agindo diretamente durante a fase de geração de padrões está embutido um outro sistema de filtro que força a montagem de padrões que estão associados unicamente com *hosts* pertencentes à uma lista previamente criada. Nesta lista estão endereços IPs de *hosts* sensíveis pertencentes à rede local monitorada ou seja, servidores de serviços eletrônicos, sistemas relacionados à segurança (*firewalls*, etc), interfaces de roteadores, etc.

Com a aplicação deste filtro muitas das associações ou seja, padrões, gerados durante a etapa anterior podem ser seguramente descartados por contribuir apenas com "ruído" na seqüência de dados (pacotes ignorados na figura 3.1.1). Assim o SDI tem seus esforços direcionados para a análise de eventos que envolvem o relacionamento local entre *hosts* sensíveis e ao relacionamento destes com *hosts* externos.

3.3 Geração de uma base normal

O uso de *data-mining* em SDIs geralmente tem seguido duas linhas. A primeira é aquela que utiliza técnicas de classificação, ou seja, é montada uma base de dados referente a ataques ou tentativas de intrusão e a partir desta base tenta-se procurar pelo que a ela é semelhante no tráfego de rede. A segunda linha utiliza técnicas de *profiling, ou seja,* é montada uma base referente ao tráfego normal e a partir dela tentar-se estabelecer uma noção do que é normal e a partir disto diferenciar o que é anomalia no tráfego de rede.

O método descrito neste artigo se caracteriza por técnicas de *profiling* [Bloe01] e a criação da base normal de padrões, também armazenada em disco no banco SQL, é feita pela captura do tráfego da própria rede sob monitoramento em períodos cíclicos de tal forma a alcançar estabilidade, ou seja, em um determinado momento não será mais necessário incluir novos padrões pois, padrões compatíveis nela já estão incluídos.

3.4 O terceiro filtro de dados – Eliminação de padrões de baixa freqüência

Ao se gerar padrões do próprio tráfego de rede não é possível ter a garantia de que traços de dados referentes a tentativas de intrusão não estarão ali presentes. Desta forma é preciso estabelecer um limiar para evitar que padrões com característica intrusiva sejam inseridos na base normal.

Seguindo o modelo de padrões do item 3.1 os campos referentes a ocorrências médias podem ser usados para a criação deste limiar a medida que se estabelece que em geral a maioria dos padrões referentes à qualquer tipo de tráfego que seja normal tem uma ocorrência média ao longo do tempo bem maior que a ocorrência média de padrões referentes à trafego anômalo pois a característica destes é de geralmente apresentar picos de ocorrências em janelas, mas no entanto apresentam baixa ocorrência média ao longo do tempo.

Usando estas características é possível definir limites mínimos e máximos de variação na taxa de ocorrência média dos padrões de modo que, na própria fase da combinação dos pacotes IP, citada no item 3.1, seja possível decidir se o padrão gerado deve ou não entrar para a linha base.

4. O Classificador: Análise e Detecção

Tendo-se em mãos uma base normal estabilizada já é possível efetuar detecção de eventos anômalos na rede por pura e simples comparação entre os padrões da base com os padrões gerados a partir do tráfego de rede, pelo mesmo processo descrito no item 3.1. Entretanto, tal processo é computacionalmente inviável dada a enorme quantidade de comparações que deverá ser feita.

Para a solução deste problema deve ser aplicado um método onde o sistema possa consultar uma estrutura simplificada que represente toda a base normal. Aqui deve ser aplicado de um programa como o RIPPER [Coh95] que é um programa de aprendizado por regras que fornece como saída um conjunto de regras do tipo "*se-então*" .

Uma observação importante é a de que quando da utilização de classificadores deve ser fornecida uma quantidade suficiente de dados para treinamento. Desta forma a base normal deve conter padrões suficientes para que durante a fase de classificação as características dos dados sejam corretamente extraídas.

Geralmente para a construção de classificadores deve-se ter disponíveis dois conjuntos de dados com características distintas (um conjunto de padrões normais e outro com padrões anormais, por exemplo). Como a linha base é composta somente por padrões normais deve ser seguido um procedimento interativo onde a cada passo a linha base é aleatoriamente dividida em dois sub-conjuntos, um para treinamento e outro para validação. Com o primeiro conjunto são geradas regras e com o segundo elas são testadas. Na próxima iteração uma nova divisão da linha base é feita e iteração se repete enquanto a taxa de erro retornada durante cada teste seja menor que um limite desejado.

Ao final do processo tem-se em mãos um conjunto de regras do classificador que representam toda a linha base e desta forma podem ser usadas para classificar o trafego. Este conjunto de regras consegue predizer como deve ser uma associação entre pacotes IP. Caso exista uma diferença entre o que foi predito e o que é encontrado no

tráfego de rede o classificador retorna um erro de classificação e os padrões são considerados como intrusivos e são destacados para posterior análise.

5. Conclusão

Quando lidando com dados brutos de rede no formato LibpCap/TCPDump, que não são formatados especificamente para propósitos específicos de detecção de intrusão, em muito é dificultada a tarefa de manualmente monitorar uma rede, em busca de eventos que caracterizem situações anômalas ou intrusivas.

Neste artigo foi mostrado que é possível a utilização de um modelo para a construção de um SDI automatizado que facilite o trabalho de monitoria aplicando pré-processamento iterativo sobre os dados brutos de rede para que dele se possa extrair as características desejáveis.

Também foi mostrado que com os três processos de filtragem citados anteriormente (seleção de conteúdo de pacotes, seleção de *hosts* sob monitoramento, eliminação de padrões de baixa freqüência) pode ser obtido um conjunto de padrões satisfatório de modo que a tarefa de classificação e aprendizado induzido trabalhe com o mínimo possível de ruído e forneça, ao final do processo de análise, um resultado satisfatório com o mínimo possível de falsos positivos.

6. Referências

[ITL0599] Computer Attacks: What They Are and How to Defend Against Them. ITL Bulletins published by the Information Technology Laboratory (ITL) of the National Institute of Standards and Technology (NIST).
URL:
http://csrc.nist.gov/publications/nistbul/05-99.pdf

[ITL1199] Acquiring and Deploying Intrusion Detection Systems . ITL Bulletins published by the Information Technology Laboratory (ITL) of the National Institute of Standards and Technology (NIST).
URL:
http://csrc.nist.gov/publications/nistbul/11-99.pdf

[Wenke99] Lee, Wenke. Mining in a Data-flow Environment: Experience in Network Intrusion Detection. Computer Science Department, Columbia University.
URL:
http://www.acm.org/pubs/articles/proceedings/ai/312129/p114-lee/p114-lee.pdf

[Fay96] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. The KDD process of extracting useful knowledge from volumes of data. Communications of the ACM, 39(11):27-34, Novembro 1996.
URL:
http://www.acm.org

[Wenke98] Wenke Lee and Sal Stolfo. ``Data Mining Approaches for Intrusion Detection" In Proceedings of the Seventh USENIX Security Symposium (SECURITY '98), San Antonio, TX, January 1998
URL:
http://www.cs.columbia.edu/~wenke/

[Bloe01] Bloedorn, Eric. Data-Mining for Network Intrusion Detection: Haw to Get Started. Mitre Technical Papers. August 2001.
URL:
http://www.mitre.org/support/papers/tech_papers_01/bloedorn_datamining/bloedorn_datamining.pdf

[Coh95] W. W. Cohen. Fast effective rule induction. In *Machine Learning: the 12th International Conference*, Lake Taho, CA, 1995. Morgan Kaufmann.