Aspectos da Arquitetura de Microcomputadores para Sistemas de Detecção de Intrusão

Horácio Hiroiti Sawame
INPE/LIT
horacio@lit.inpe.br

Stephan Stephany
INPE/LAC
stephan@lac.inpe.br

Airam Jônatas Preto
INPE/LAC
airam@lac.inpe.br

Resumo

Uma importante maneira de assegurar um bom nível de segurança em um sistema de informações baseado em rede local é utilizar um Sistema de Detecção de Intrusão de Rede - SDIR, cuja finalidade é detectar atividades hostis e/ou maliciosas, que tenta analisar todos os pacotes de dados transmitido na rede. Como exemplo da primeira categoria há os ataques de Negação de Serviço, enquanto no segundo, varredura de portas. Um SDIR é geralmente instalado num computador padrão de arquitetura IA-32 e sistema operacional GNU/Linux. Para que se possa capturar todos os pacotes e processá-los em tempo real, levando-se em conta redes de alta velocidade, é importante estar familiarizado com a arquitetura e tecnologia dos computadores, incluindo hardware *(sub-sistemas de entrada/saída, barramentos internos de interligação, etc) e* software *(*drivers *de dispositivos, bibliotecas, sistema operacional).*

Abstract

An important way to ensure a good security level in a network based information system is to use a Network Intrusion Detection System - NIDS, aiming at the detection of hostile and/or suspicious activities, that try to analyze every packet transmitted in the network. As an example of the first category there are Deny of Service attacks, while in the second, port scanning. A NIDS is commonly based on IA-32 standard computers running GNU/Linux. In order to capture and process every packet in near real time, considering the standard high speed networks, it is important to be acquainted with the computer architecture and technology, including hardware (I/O subsytem, internal busses, etc.) and software (drivers, libraries, operating systems) issues.

1. Introdução

A tecnologia de redes locais tem evoluído bastante e tem encontrado grande receptividade por parte das organizações modernas. Muitas dessas redes são conectadas à Internet, provendo os mais diversos serviços. As vantagens de se ter maior conectividade tem como contrapartida maiores riscos de segurança devido às vulnerabilidades nos sistema e/ou nas deficiências dos protocolos em uso que vão sendo descobertas a cada dia. Muitas dessas vulnerabilidades são percebidas pelos administradores dos sistemas somente algum tempo após a divulgação de ferramentas que exploram essas vulnerabilidades nas comunidades *hacker* ou nas comunidades ligadas à segurança de sistemas.

A crescente preocupação das empresas com a segurança de suas redes e de seus sistemas conectados à Internet tem aumentado a demanda por sistemas de controle de acesso (filtros de pacote ou *firewalls e proxies*) e de monitoração de tráfego (*sniffers* e detectores de intrusão de rede). O controle de acesso permite evitar que entidades externas tenham acesso aos sistemas internos das empresas e ao mesmo tempo possibilita a limitação de acesso dos usuários internos aos recursos externos que interessam à empresa. A monitoração de tráfego permite que se detecte indícios de atividades ilícitas tais como varreduras de portas, disseminação de vírus e vermes de rede, ataques para desabilitação de serviços (DOS, *Denial Of Service*).

Um componente importante (porém não o único, deve sempre fazer parte de uma política de segurança mais global) é o sistema de detecção de intrusão de rede (SDIR), que deve monitorar todo o tráfego da rede (em geral o tráfego baseado no protocolo TCP/IP), vistoriando o cabeçalho do pacote (que contém informações sobre o protocolo e portas utilizados, endereços IP de origem e destino) e seu conteúdo, confrontando o com regras e heurísticas que podem determinar se esse pacote é lícito ou pode fazer parte de alguma atividade hostil.

No caso do Snort, uma ferramenta de detecção de intrusão de rede de código livre, houve uma mudança arquitetural em sua versão 2.0 (lançado em abril de 2003) em que as suas regras são agrupadas em conjuntos e tratadas por algoritmos mais apropriados para procura de seqüências de caracteres. Além disso sua estrutura é bastante modular, permitindo agregar diversos filtros de

pré-processamento (como reconstrução de sessão) e processadores de saída, permitindo saída direta para sistemas gerenciadores de bancos de dados, como MySQL (`http://www.mysql.org`) e PostgreSQL (`http://www.postgresql.org`).

Apesar desses aperfeiçoamentos, a medida que a velocidade das redes aumenta, a ocorrência de perda de pacotes (a possibilidade de não se detectar um evento, ou falso negativo) tenderá a aumentar e para compensar o Snort demandará uma capacidade computacional cada vez maior, talvez além do disponível em equipamentos comuns.

O que se propõe a fazer é identificar gargalos no *hardware* e no *software* e propor alterações que permitam a plena captura de pacotes em redes de alta velocidade evitando o uso de equipamentos e componentes especializados e caros.

2. Processamento de Pacotes

Os pacotes que passam pela rede são capturados por uma placa de rede (o SDIR deve estar posicionado num ponto da rede em que o todo o tráfego possa ser monitorado, como por exemplo na sub-rede de entrada do *firewall*) em modo promíscuo (a placa de rede deve receber todos os pacotes que passam, independemente de sua origem ou destino). A placa de rede executa algumas verificações preliminares e descarta pacotes inválidos (CRC incorreto, frames truncados). Dependendo do tipo de barramento em que a placa de rede está conectada ela inicia um processo de transferência do pacote capturado para a memória principal do sistema (*buffer* da placa de rede) e sinaliza para o seu *driver* de dispositivo que um pacote chegou. Em microcomputadores PC atuais com barramentos PCI, a placa de rede inicia um processo de transferência *bus-mastering*, ou seja toma o controle do barramento, transfere o pacote da memória local da placa de rede para a memória principal (*buffer* da placa de rede, vide figura 2), libera o controle do barramento e gera uma interrupção de *hardware*, que será atendida pelo APIC (Controlador de Interrupção Programável Avançado) do microcomputador. O APIC, por sua vez, ativa a rotina de manipulação de interrupções do sistema operacional, e isto leva à execução da Rotina de Atendimento da Interrupção do *driver* de dispositivo da placa de rede.

Para permitir que aplicações no espaço de usuário possam se utilizar dos pacotes capturados (até aqui os pacotes foram processados no espaço de *kernel*) bibliotecas como BPF (*Berkeley Packet Filter*, para sistemas operacionais derivados do BSD), *libpcap* (`http://www.tcpdump.org/`, para sistemas operacionais UNIX como Sun Solaris, HP HP-UX e clones como Linux) e *WinPcap* (`http://winpcap.polito.it/`, para o sistema operacional Microsoft Windows) foram criadas. Muitos programas de monitoramento de rede empregam estas bibliotecas. Funções como *tap()* podem ser associadas à Rotina de Atendimento

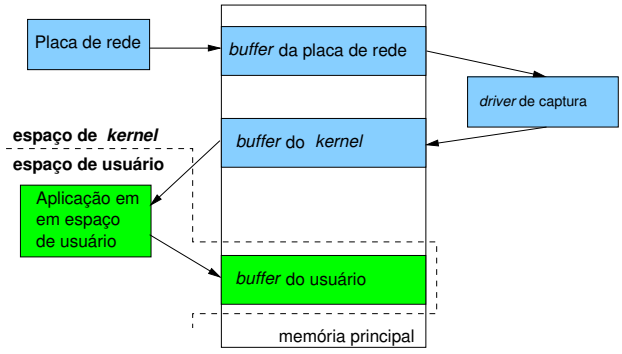

Figura 1: trajetória do pacote, da placa de rede à aplicação.

da Interrupção da placa de rede, permitindo que a aplicação do usuário defina regras de filtragem (seleção de protocolo ou de determinados campos do cabeçalho Ethernet). Após a filtragem os pacotes aceitos são associados com informações da camada física (comprimento e horário da recepção) e são copiados no *buffer* do *kernel*. A aplicação do usuário então pode copiar o pacote para o *buffer* de usuário por uma função *read()* .

3. Gargalos

3.1. Forma de Transmissão dos Pacotes

A natureza serial da transmissão dos pacotes impõe um atraso pois é necessário aguardar recebimento do último bit de cada palavra que se possa processar verificações de CRC, identificar truncamentos e finalmente enviar o conteúdo do buffer para a memória do microcomputador.

3.2. Placas de Interconexão de Rede

Na referência [1], os autores, utilizando manipulação dos MSR (registradores específicos de cada família de processadores), bibliotecas dinâmicas especialmente desenvolvidas para medir o número de ciclos de CPU utilizados num trecho de código e o pacote *Intel Vtune Performance Analyzer*, levantaram o perfil de execução (quantificando o custo computacional em termos de ciclos de CPU) da biblioteca WinPcap e dos *drivers* de duas placas de rede de alto desempenho durante a captura de pacotes de rede.

As seguintes conclusões foram apresentadas:

- As placas de rede e seus *drivers* de dispositivo (pelo menos os modelos testados pelos autores de [1]) requerem pouco esforço da CPU. Entretanto o número de interrupções a serem atendidas (o que influencia no custo do sistema operacional em atendê-las) e o número de operações de entrada/saída para acessar os

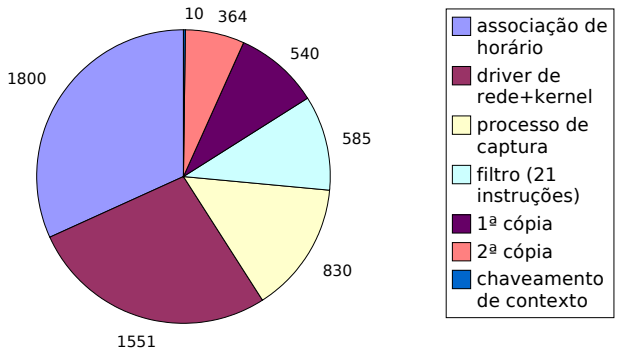

Figura 2: divisão dos ciclos de CPU por atividades

registradores da placa de rede (o que influencia no custo do *driver* de dispositivo) tem um grande impacto no desempenho da captura de pacotes. Em placas de rede mais modernas é possível transferir mais de um pacote por interrupção, o que proporciona um melhor desempenho.

- Durante a transferência do tipo PCI *bus-mastering* dos pacotes da placa de rede para a memória principal, o uso de CPU é mínimo mas se a quantidade de dados for muito grande o barramento não pode ser utilizado pela CPU o que pode afetar o desempenho do sistema devido a atrasos no atendimento de requisições da CPU.

- A análise quantitativa de custo de todos os processos envolvidos na captura de pacotes mostrou que grande parte do custo envolve *hardware* e *driver* de dispositivo (59% dos ciclos de CPU são relativos à marcação do horário de captura e ao processamento do *driver* da placa de rede e do processamento das interrupções pelo *kernel*, ver figura 3.2), o que inviabiliza otimizações pois o *hardware* em geral não pode ser reprogramado e o fabricante em geral não libera o código do *driver* (no caso de versões para MS Windows). Uma solução seria utilizar placas mais caras e que implementam em *hardware* algumas das operações efetuadas tradicionalmente pela biblioteca de captura. Um exemplo é a família de placas DAG da Endace (`http://www.endace.com/`), que já faz a associação dos pacotes capturado com o horário de captura na própria placa de rede.

- A utilização de instruções e registradores específicos da arquitetura da CPU na otimização da biblioteca WinPcap permitiu a redução do custo de processamento em cerca da metade. Entretanto a portabilidade da biblioteca fica muito prejudicada, pois essas instruções e registradores não são padronizadas, podendo nem existir numa determinada família de microprocessador.

3.3. Barramentos

O barramento para interconexão de periféricos PCI (*Peripheral Component Interconnect*) foi proposto pela Intel (`http://www.intel.com/`) em 1992 é a principal forma de comunicação entre o microprocessador e seus periféricos, entre os quais estão as placas de conexão à rede local, e admite transferências de palavras de 32 bits na freqüência de 33 MHz (a versão mais comumente encontrada nos microcomputadores de mesa, com taxa de transferência de 132 Mbyte/s) até versões de 64 bits na freqüência de 66 MHz (somente encontrada em alguns servidores mais caros, com taxa de transferência de 528 Mbyte/s).

A taxa de transferência das placas de rede *gigabit* é de 125 Mbyte/s (máximo teórico) pode se perceber que o tráfego de uma só placa de rede já poderia causar uma saturação do barramento na maioria dos microcomputadores de mesa. Como a utilização de placas de rede *gigabit* ficaria limitada a equipamentos mais caros a solução apontada pela indústria foi a de lançar uma nova versão do barramento em que a transferência se dá principalmente por ligações seriais ponto-a-ponto, a PCI-Express (anteriormente conhecida como 3GIO *3rd Generation I/O*). A interconexão entre a CPU e os periféricos será feita por intermédio de um comutador em vez de por barramento, analogamente às redes locais, onde um comutador (ou *switch*) permite aumentar o desempenho de uma rede de par trançado em relação a equipamentos de compartilhamento como *hubs* ou redes de cabo coaxial. Como o número de via paralelas diminui, o traçado de circuito impresso das diversas placas pode ser mais simplificado e tanto a interferência eletromagnética entre os componentes como o custo de produção tendem a cair, apesar do aumento de freqüência de operação. Entretanto deverão existir sistemas com ambos os barramento (PCI e PCI-Express, para placas de vídeo e de rede *gigabit*) por um longo período, a exemplo dos sistemas atuais ainda com os barramentos PCI e ISA.

3.4. Outros Barramentos Internos

Outro fator é a interface de conexão do processador (em muitos sistemas conhecido como FSB, *Front Side Bus*). Nos modelos atuais do processador AMD Athlon XP a freqüência de operação do FSB chega a 166 MHz em *Double Data Rating* (DDR), ou popularmente “333 MHz DDR”, resultando numa taxa de transferência de 2,6 Gbyte/s. No Athlon XP de 3,2 GHz (provavelmente o último modelo dessa família) o FSB opera a 200 MHz em DDR (“400 MHz”), resultando numa taxa de transferência de 3.2 Gbyte/s. Nos processadores Intel Pentium 4 da série C o FSB opera em quatro canais de 100 MHz DDR (“800 MHz” segundo o *marketing* da empresa), com taxa de transferência de 6,4 Gbyte/s. Como a Intel praticamente abando-

nou a tecnologia RAMBUS (tecnologia desenvolvida pela empresa de mesmo nome e que devido ao seu alto custo inicial foi sendo preterida pelo mercado) de memória em favor da tecnologia SDRAM DDR (*Synchronous Dynamic Random Access Memory*), tornou-se necessário chegar a soluções de controlador de memória de canal duplo como nos controladores baseados na famílias Intel 7205, Intel 865PE e Intel 875, o que possibilita que dois subsistemas de memória forneçam uma taxa de transferência de 3,2 Gbyte/s cada, em paralelo, alimentando o FSB do processador.

A nova família de processadores da AMD marcam uma transição para processadores de 64 bits para uso geral (os processadores de 64 bits da Intel, o Itanium, focalizam o mercado de servidores). Os processadores Athlon 64 e Athlon 64 FX incorporam controladores de memória (nos projetos tradicionais o controlador de memória fica na placa-mãe e não no processador) e a interligação dos processadores com a placa mãe e entre si (no caso das versões multiprocessadas) é feita pelo barramento *HyperTransport* (desenvolvido pela AMD em conjunto com outras empresas como a nVidia, SiS, Transmeta, Apple, SGI, Sun, Sun, VIA, Toshiba, HP, Tektronix, NEC ALi, Cisco e outros).

O processador Intel Pentium 4 da série C foi introduzido o *HyperThreading* (antes presente somente na linha de processadores Xeon, para servidores), recurso que permite utilizar as unidades de processamento ociosas da arquitetura *pipeline* como um segundo processador. Isto permite que aplicações projetadas para aproveitar multiprocessamento (desde que o sistema operacional e o hardware habilitem o multiprocessamento) possam usar estes processadores como se fossem dois. Isto beneficiaria aplicações projetadas para utilizar processamento paralelo simétrico, o que, infelizmente, não é o caso da maioria das aplicações que utilizam bibliotecas de captura de pacotes.

Os processadores AMD Opteron/Athlon 64/Athlon 64FX, possuindo um conjunto de instruções compatível com o da arquitetura Intel de 32 bits, permitem executar a maioria da aplicações sem recompilação e ainda com um ganho devido ao barramento de dados mais largo (128 bits). Isto possibilitaria aumento de desempenho das aplicações de maneira geral pelo simples fato de se trabalhar com 64 bits.

3.5. Armazenamento

Quando se torna necessário armazenar grandes volumes de dados, o que é o caso do Snort configurado para armazenar os registros (*logs*) e os pacotes de rede suspeitos (*dump* em formato *tcpdump*) no próprio sistema, a comunicação com os discos rígidos passam a ter uma grande importância. Aqui novamente o que se observa a tendência de se utilizar protocolos serias em vez de paralelos. As interfaces Serial-ATA (`http://www.serialata.org/`) em sua primeira versão implementam taxas de transferência de 150 Mbyte/s. As características novas são:

- Custo compatíveis com equipamentos de mesa;
- Conexão e desconexão “à quente”;
- Cabos mais estreitos permitindo melhor circulação de ar nos gabinetes;
- Conexão ponto-a-ponto evitando problemas de configuração e terminação;
- Escalabilidade.

Ainda que as interfaces sejam bastante rápidas, os discos rígidos, devido à sua natureza mecânica têm limitações na taxa de transferência (devido a leitura da mídia magnética, a velocidade de rotação é um dos parâmetros mais importantes) e no tempo de acesso (influenciado pelo projeto dos atuadores dos cabeçotes de leitura e pela velocidade de rotação). Para tentar aumentar o desempenho, soluções como configuração de diversos discos em RAID (*Redundant Arrays of Independent Disks*, antigamente *Redundant Arrays of Inexpensive Disks*) são utilizados. Configurações como RAID 5 (3 ou mais discos com as palavras de redundância distribuídas entre eles) ou 1+0 (as palavras são distribuídas entre metade dos discos, enquanto a outra metade espelha a primeira) oferecem melhor desempenho nas taxas de transferência e tolerância à falha.

Mesmo assim devido à possibilidade de conflito no compartilhamento do barramento PCI com as placas de rede seria aconselhável fazer o armazenamento dos dados (de preferência comprimidos) em um outro sistema.

4. Conclusões

Há uma necessidade constante de se manter atualizado com relação a arquitetura e tecnologia de *hardware* pois o mercado de microcomputadores é muito dinâmico e freqüentemente surgem tecnologias que tem bastante impacto no desempenho das aplicações de captura de pacotes de rede. Além disso é necessária uma análise muito criteriosa para se determinar os reais gargalos de uma aplicação, como ficou demonstrado na referência [1], para que se possa balancear os custos de implantação. Utilização de placas especiais, mais caras, podem ser uma solução em alguns casos, otimizações específicas para determinadas arquiteturas podem ser indicadas para outros casos.

As próximas etapas a serem desenvolvidas são:

- determinar para o sistema operacional Linux, os *drivers* padrões das placas de rede disponíveis no INPE e a biblioteca *libpcap* os tempos de execução dos diversos processos, similarmente ao trabalho da referência [1];

- verificar a possibilidade de otimizar o *driver* de uma placa de rede, pois o código fonte está disponível ao contrário da maioria dos *drivers* para o sistema operacional MS Windows;

- verificar a possibilidade de otimizar a biblioteca *libpcap* para uma arquiteura específica de processador;

- tentar adquirir placas especializadas para aquisição de pacotes e determinar o impacto de seu uso nas aplicações;

- modificar o código do Snort para trabalhar com múltiplos processos e/ou threads, para avaliar a viabilidade de se utilizar uma arquitetura de multicomputadores;

Referências

[1] Loris Degiovani, Mario Baldi, Fulvi Risso, and Gianluca Varenni. Profiling and optimization of software-based network-analysis applications. In *Symposium on Computer Architecture and High Performance Computing*, pages 226–34. Dipartimento di Automatica e Informatica, Politecnico di Torino, IEEE Computer Society, 2003.