Detecção de Ataques pela Constatação de Violação dos Protocolos IP e TCP

Norma Rodrigues Gomes[1], Luiz Antonio da Frota Mattos[1]

[1]Departamento de Ciência da Computação – Universidade de Brasília (UnB)
Campus Universitário Darcy Ribeiro – 70.910-900 – Brasília – DF – Brazil
norma.nrg@dpf.gov.br, frota@unb.br

Abstract. *One of the big challenges in network intrusion detection´s area is the limitation imposed by the use of well-known attacks signatures, disabling the previous detection of new attacks. This work presents a packet analysis methodology whose purpose is to detect anomalous behaviors, not basing on attacks signatures but verifying if the network protocols are not being violated, basing on the content of the respective headers. The biggest benefit is the possibility of anomalies or inadequate behaviors detection, that can correspond, total or partially, to variations of well-known attacks and even unknown.*

Resumo. *Um dos grandes desafios na área de detecção de intrusão em redes de computadores é a limitação imposta pelo uso de assinaturas de ataques conhecidos, incapacitando a detecção prévia de novos ataques. Este trabalho apresenta uma metodologia de análise de pacotes cuja finalidade é detectar comportamentos anômalos, não com base em assinaturas de ataques e sim verificando se os protocolos de rede não estão sendo violados, com base no conteúdo dos respectivos cabeçalhos. O maior benefício é a possibilidade de detecção de anomalias ou comportamentos inadequados, que podem corresponder, total ou parcialmente, a variações de ataques conhecidos e até mesmo desconhecidos.*

1. Introdução

As redes de computadores proporcionaram um grande avanço na área tecnológica, principalmente no tocante à facilidade de troca de informações, o que se acentuou ainda mais com a chegada da Internet. Entretanto, junto com este avanço, vieram também problemas relacionados a questões de segurança, de forma que manter uma rede protegida visando evitar incidentes que causem prejuízos, tornou-se uma preocupação constante, levando à implementação de diversos mecanismos de segurança.

Um mecanismo de segurança que vem ganhando cada vez mais espaço é denominado Sistema de Detecção de Intrusão - SDI. A maioria dos SDIs baseia-se em assinaturas de ataques, as quais, basicamente, descrevem padrões de comportamento conhecidos, considerados suspeitos e que constituem problemas de segurança [Amoroso, 1999]. Entretanto, um dos grandes desafios na área de detecção de intrusão é a limitação imposta pelo uso de assinaturas de ataques conhecidos, incapacitando a detecção prévia de novos ataques com assinaturas desconhecidas [Krügel et al., 2002].

Nossa proposta consiste em realizar uma análise de pacotes cuja finalidade é detectar a existência de comportamentos anômalos, mas não com base em assinaturas de

ataques e sim verificando se os protocolos utilizados em um dada rede não estão sendo violados, com base no conteúdo dos respectivos cabeçalhos. Com esse tipo de análise, além de ser possível detectar alguns tipos de ataques conhecidos, incluindo até mesmo possíveis variações destes, novos ataques, que façam uso de alguma violação de protocolo ainda não utilizada em nenhum ataque conhecido, poderiam ser detectados.

Neste artigo será apresentada uma metodologia de análise de pacotes, visando detecção de ataques, baseada na especificação dos protocolos IP e TCP. A Seção 2 apresenta informações sobre o problema causado por pacotes que violam protocolos de rede e introduz a análise de pacotes proposta. A Seção 3 mostra a consistência da nossa proposta, definindo a linha de raciocínio e os métodos empregados. A Seção 4 compara a abordagem de análise adotada, que se refere à violação de protocolos, com o método baseado em assinaturas de ataques. A Seção 5 resume as conclusões obtidas e apresenta caminhos futuros para aprimoramento do trabalho proposto.

2. Descrição do Problema

Em uma rede de computadores, a interação entre os diversos hosts que a compõem ocorre através do uso de protocolos, os quais devem ser seguidos para que todos os envolvidos na comunicação "falem" a mesma língua, possibilitando, dessa forma, a troca de dados. Assim, quando se tem uma rede que trabalha com protocolos TCP/IP, por exemplo, o que se espera é que todos os pacotes que trafeguem pela rede atendam às especificações destes protocolos.

Entretanto, é possível montar pacotes que violem os protocolos de comunicação, causando reações inesperadas no lado receptor dos pacotes, podendo configurar, inclusive, uma forma de ataque. Uma rede de computadores está vulnerável a sofrer diversos tipos de ataques com múltiplos propósitos, tais como: fazer reconhecimento da rede (scanning de rede) para identificar quais serviços ela oferece ou qual o sistema operacional utilizado, interromper ou negar acesso de usuários legítimos a serviços, servidores ou outros recursos (negação de serviço ou DoS – Denial of Service), dentre outros.

Mecanismos de segurança têm sido desenvolvidos para detectar ataques a redes de computadores, dentre os quais destaca-se o SDI (Sistema de Detecção de Intrusão). A grande maioria dos SDIs baseia-se em assinaturas de ataques. Na prática, o que acontece é que, uma vez ocorrido o ataque, o seu comportamento é analisado e é gerada uma espécie de regra que traduz esse comportamento, de forma que, sempre que esse comportamento ocorrer o respectivo ataque é detectado.

Entretanto, os SDIs que contam com comparação de padrões de comportamento que representam assinaturas de ataques conhecidos, são incapazes de detectar previamente ataques não vistos com assinaturas diferentes, o que representa um dos grandes desafios na área de detecção de intrusão. Uma abordagem que vale a pena ser trabalhada, de acordo com [Allen et al., 2000], é ter uma noção do que constitui comportamento de sistema normal e detectar divergências a partir deste. Tal abordagem para detectar ataques desconhecidos consiste em determinar o que representa comportamento normal dentro de redes, *hosts* ou aplicações, e sinalizar atividade que não reflete o que é esperado.

Ao observar as assinaturas de ataques, nota-se que algumas delas representam, na verdade, violações aos protocolos de rede (por exemplo, TCP/IP), que deveriam ser

seguidos. Assim, é importante entender como é o comportamento padrão dos protocolos envolvidos no tráfego de dados de uma rede, a fim de detectar se o mesmo atende às especificações do protocolo em questão.

Essas especificações podem ser obtidas em documentos denominados de RFC (Request For Comments), que descrevem padrões esperados para protocolos individuais [Northcutt et al., 2000]. Os documentos RFCs são publicados e mantidos pelo IESG (Internet Engineering Steering Group) após serem amplamente discutidos e testados na comunidade, através de listas de discussão mantidas pelo IETF (Internet Engineering Task Force).

Portanto, ao verificar o protocolo que está sendo utilizado em uma dada rede, bem como suas respectivas especificações, definidas em RFCs, alguns comportamentos anômalos podem ser detectados, tornando, inclusive, desnecessária a existência de algumas assinaturas de ataques e possibilitando a detecção de algumas variações de assinaturas existentes.

3. Análise de Pacotes para Detecção de Violação de Protocolos

Nossa proposta consiste em realizar uma análise de pacotes cuja finalidade é detectar a existência de um comportamento divergente daquele esperado e definido pelo protocolo que estiver sendo utilizado. Para tanto definimos o escopo de trabalho, em termos de protocolos e tipos de análise, métodos para formalizar as regras de comportamento esperado de um dado protocolo, e uma implementação desse analisador de pacotes, em nível de protótipo, conforme detalhado a seguir.

3.1. Escopo da Análise de Pacotes

Para fins da análise a ser realizada serão verificados os pacotes que trafegam em redes TCP/IP, visto que o TCP/IP é o protocolo mais usado em redes locais, o que se deve, basicamente, à popularização da Internet. Mais precisamente, restringiremos a nossa análise aos pacotes que usam os protocolos IP (na camada de rede) e TCP (na camada de transporte), de forma que a análise é feita em cima dos cabeçalhos IP e TCP. Tal restrição é obtida através de um filtro, ao qual a rede a ser analisada é submetida, utilizando-se a ferramenta *windump* (http://windump.polito.it) para realização da coleta de pacotes.

Visto que trabalharemos com TCP, que é um protocolo confiável orientado a conexão, a análise dos pacotes será feita em duas etapas, a saber:

- Etapa 1 – Análise sem Estado (Stateless Inspection)
- Etapa 2 – Análise com Estado (Stateful Inspection)

Na Análise sem Estado é feito o exame de pacotes individuais, ou seja, a análise fica restrita ao conteúdo dos campos dos cabeçalhos de um pacote em questão. O nome “sem estado” é devido ao fato de que, neste tipo de análise, não se mantém uma trilha do estado da conexão, isto é, ao receber um pacote para ser analisado não se tem idéia se o mesmo é parte de uma conexão existente ou não.

Já na Análise com Estado é feito o exame de uma seqüência de pacotes, ou seja, a análise engloba dados de cabeçalhos de mais de um pacote, permitindo detectar padrões de comportamento entre seqüências de pacotes e realizar correlações. Neste tipo

de análise são mantidas informações sobre o estado das conexões estabelecidas, permitindo saber quais pacotes fazem parte de uma dada conexão.

3.2. Análise sem Estado

A primeira etapa da análise dos cabeçalhos TCP/IP será feita em dois termos:

a) em termo dos possíveis formatos dos cabeçalhos; e
b) em termo das regras que impõem algumas restrições de valores aos campos dos mesmos, a fim de que o cabeçalho seja significativo.

Ao verificarmos se um dado formato de cabeçalho é permitido estamos fazendo, na realidade, uma análise sintática do mesmo. Assim, na análise sintática será verificada a estrutura do cabeçalho, que possui como unidade básica de formação os campos. O cabeçalho será considerado sintaticamente válido se o mesmo obedecer às regras sintáticas, as quais determinam quais cadeias de campos podem formar cabeçalhos (regras de formação).

Já ao aplicarmos regras com restrições sobre os valores dos campos do cabeçalho, a fim de que o mesmo seja significativo, estamos fazendo uma análise semântica. Na análise semântica será verificado o significado do cabeçalho em termos dos valores dos seus campos. Analogamente, o cabeçalho será considerado semanticamente válido se o mesmo obedecer às regras semânticas, as quais determinam se o cabeçalho possui significado com base nos valores presentes em seus campos.

Portanto, para que o cabeçalho seja considerado válido é preciso que o mesmo o seja sintática e semanticamente. Iniciaremos nossa análise pela parte sintática, conforme segue.

3.2.1. Análise Sintática

Na análise sintática, o primeiro passo a ser dado consiste em definir o formato dos dados com os quais vamos trabalhar, para então serem definidas as regras sintáticas.

Os valores dos campos dos cabeçalhos IP e TCP serão expressos em termos de dígitos hexadecimais. Alguns campos possuem regras de formação específica, limitando a faixa de valores permitida nos mesmos. Tais regras foram identificadas através do estudo dos documentos RFCs e são parte integrante da análise sintática realizada.

As regras sintáticas dos cabeçalhos foram definidas e formalizadas através do uso da metalinguagem denominada BNF (Backus-Naur Form). O conjunto de todas as construções sintáticas referentes aos cabeçalhos IP e TCP, que correspondem às respectivas regras sintáticas, serão referenciadas como componente BNF [Sager, 1981].

A Figura 1 mostra exemplos de algumas construções sintáticas que compõem o componente BNF do cabeçalho IP.

<Cab_IP> ::= <Versao> <Tam_Cab_IP> <Tip_Serv> <Tam_Tot_Dtg> <Id> <Flags-Offset> <TTL> <Protocolo> <Checksum_IP> <End_Origem> <End_Destino> | <Versao> <Tam_Cab_IP> <Tip_Serv> <Tam_Tot_Dtg> <Id> <Flags-Offset> <TTL> <Protocolo> <Checksum_IP> <End_Origem> <End_Destino> <Opcoes-Pad_IP>.

<Versao> ::= 4.

<Tam_Cab_IP> ::= 5|6|7|8|9|a|b|c|d|e|f. (i.e., <Tam_Cab_IP> >= 5 palavras de 32 bits (5 hex))

Figura 1. Parte do Componente BNF do Cabeçalho IP

A primeira definição BNF, mostrada na Figura 1, é uma seqüência de duas opções separadas pela linha vertical, onde cada opção apresenta um conjunto possível de campos do cabeçalho IP, de acordo com a RFC 791. Analogamente, cada campo é definido individualmente, apresentando os respectivos conjuntos de valores aceitos. Comentários colocados entre parênteses têm o intuito de esclarecer o significado de algumas regras.

3.2.2. Análise Semântica

A partir do estudo e análise do conteúdo dos documentos RFCs (RFC 791, 793, 1323 e 2018) e de algumas considerações feitas acerca destes documentos em [Northcutt et al., 2001], foram extraídas as regras semânticas, além daquelas definidas nos componentes BNF, que definem as relações que devem existir entre os valores dos campos dos cabeçalhos a fim de que os mesmos sejam válidos semanticamente.

A Figura 2 mostra exemplos de algumas regras semânticas referentes aos cabeçalhos IP e TCP.

a) <Tam_Tot_Dtg> deve ser maior ou igual ao (<Tam_Cab_IP> *4)

b) <End_Origem> deve ser diferente de <End_Destino>, para que os mesmos façam sentido.

c) Bit ACK = 1 (campo <Flags_TCP>) se e somente se <Num_Ack> é maior que 1.

Figura 2. Exemplos de Regras Semânticas para os Cabeçalhos IP e TCP

As duas primeiras regras, mostradas na Figura 2, referem-se ao cabeçalho IP. A primeira trata dos campos <Tam_Tot_Dtg> (tamanho total do datagrama) e <Tam_Cab_IP> (tamanho do cabeçalho IP), enquanto a segunda trata dos endereços de origem e destino. Já a terceira regra refere-se ao cabeçalho TCP e trata do flag ACK e do <Num_Ack> (número de *acknowledgment*).

Além das regras sintáticas (componentes BNF) e semânticas (conjunto de restrições dos valores dos campos dos cabeçalhos), já citadas, vale destacar que a questão da fragmentação de pacotes também é tratada na nossa análise, visto que o processo de fragmentação também implica em cumprimento de regras, as quais são estabelecidas pelo protocolo IP. Estas regras tratam, basicamente, de questões referentes ao tamanho do fragmento e ao respectivo posicionamento quando do processo de remontagem.

Após a definição das regras dos cabeçalhos, contabilizadas num total de trinta e uma regras, o próximo passo consiste na aplicação de um modelo lógico para criar uma teoria que corresponde ao conjunto de regras que devem ser satisfeitas para que os cabeçalhos IP e TCP sejam considerados válidos.

3.3. Aplicação de um Modelo Lógico para Formalização das Regras dos Cabeçalhos

A fim de definir logicamente as regras sintáticas e semânticas estabelecidas para os cabeçalhos IP e TCP, é construído um formalismo aplicando-se a lógica de primeira ordem.

As regras dos cabeçalhos IP e TCP são interpretadas a partir da construção de uma teoria de primeira ordem. Essa teoria é formada por um conjunto de fórmulas da lógica de primeira ordem gerado a partir das regras sintáticas e semânticas definidas

para os cabeçalhos. A Tabela 1 exemplifica a geração de fórmulas a partir das regras pré-definidas.

Tabela 1. Exemplo de Geração de Fórmulas a partir das Regras dos Cabeçalhos.

Regra	Fórmula
<Tam_Cab_IP> ::= 5|6|7|8|9|a|b|c|d|e|f.	$\forall x$ (tam_cab_ip(x) $\leftarrow$ x >= 5)
<End_Origem> é diferente de <End_Destino>	$\forall x \forall y$(end_origem_end_destino(x,y) $\leftarrow$ x <> y)

A primeira fórmula da Tabela 1 deve ser lida da seguinte forma: "para todo x, x é um tamanho de cabeçalho IP válido se x é maior ou igual a 5". Analogamente, a segunda fórmula quer dizer: "para todo x, para todo y, x é um endereço de origem válido em relação a um endereço de destino y se x é diferente de y".

Após a formalização das regras sintáticas e semânticas dos cabeçalhos, é definida, então, a fórmula principal, a qual utiliza as fórmulas geradas anteriormente para definir o que é um pacote com cabeçalhos válidos, da seguinte forma: "$\forall x \forall y$ (pacote_cab_val(x,y) $\leftarrow$ regras_ok(x,y))". Tal fórmula possui o seguinte significado: "x e y são, respectivamente, cabeçalhos IP e TCP válidos de um pacote se x e y atendem a todas as regras". A expressão "regras_ok(x,y)" será verdadeira se a conjunção de todas as fórmulas correspondentes às regras dos cabeçalhos forem verdadeiras.

Assim, o predicado "pacote_cab_val" será utilizado para consultar se um dado conjunto de valores representa um pacote com cabeçalhos IP e TCP válidos.

3.4. Protótipo para Realização da Análise sem Estado (RECAB)

A implementação das regras que definem a validade dos cabeçalhos IP e TCP, formalizadas através de um conjunto de fórmulas da lógica de primeira ordem, é realizada utilizando-se programação em lógica.

Uma das grandes vantagens da programação em lógica, em relação à programação convencional, é que a tarefa do programador resume-se, praticamente, à especificação do problema que deve ser solucionado, visto que as linguagens lógicas podem ser vistas simultaneamente como linguagens para especificação formal e linguagens para a programação de computadores. Portanto, a teoria criada para formalizar as regras dos cabeçalhos, a qual consiste de um conjunto de fórmulas da lógica de primeira ordem, na realidade, corresponde a um programa em lógica.

Assim, o protótipo para testar as regras referentes aos cabeçalhos dos protocolos IP e TCP, denominado de RECAB, foi desenvolvido em linguagem Prolog. A essência do RECAB consiste na análise de pacotes individuais onde são verificadas se as regras referentes aos protocolos IP e TCP são obedecidas. Caso haja alguma violação de protocolo o pacote é identificado e os erros detectados são listados. Para tanto, é feita a introdução de uma variável denominada "resultado" em cada pacote analisado, a qual conterá uma lista com indicação das regras que não forem satisfeitas, correspondendo, na realidade, ao resultado da análise.

A fim de que os pacotes, que se encontram no formato *windump*, sejam processados pelo protótipo, é feita uma conversão dos mesmos em termos Prolog, de forma que os pacotes passam a ser representados por fórmulas atômicas [Casanova,

1987]. A idéia original, referente ao processamento dos pacotes pelo RECAB, consistia em colocar os pacotes sob análise como parte do próprio programa Prolog.

Entretanto, a fim de que o programa não ficasse sobrecarregado, comprometendo sua execução, devido ao grande número de pacotes que foram analisados, os pacotes, definidos como termos Prolog, foram armazenados em um arquivo texto. Este arquivo, cujo conteúdo corresponde ao tráfego de rede coletado para análise, é passado como parâmetro de entrada para o RECAB, o qual faz uma leitura seqüencial do mesmo, analisando cada pacote lido, e relatando caso haja alguma violação de protocolo, até que o final do arquivo seja atingido.

3.5. Análise com Estado

Da mesma forma que na análise sem estado, o escopo da análise com estado são os cabeçalhos IP e TCP, tendo como ênfase a conexão TCP. De acordo com [Northcutt, 2000], os comportamentos mais genéricos que se pode analisar, quando o escopo é o protocolo TCP, referem-se a:

1) Estabelecimento da Conexão;
2) Transferência de Dados; e
3) Finalização da Conexão.

Seguindo esta linha base, nesta etapa que se refere à análise com estado, é proposto um esquema para estudo do comportamento da conexão TCP/IP, o ESTCON, onde será verificado se uma conexão foi estabelecida de acordo com o processo de *three-way handshake*, se houve transferência de dados após o estabelecimento da conexão, e, por fim, se a conexão foi finalizada através do uso do flag FIN ou RESET.

O ESTCON tem como foco estudar o comportamento de uma dada seqüência de pacotes, a fim de identificar quando uma situação de violação do comportamento esperado do protocolo representa algum tipo de ataque.

A implementação do ESTCON apóia-se na utilização de um banco de dados para realizar correlação de pacotes. O arquivo texto contendo os pacotes representados por termos Prolog, utilizado na análise sem estado, foi importado para o Banco de Dados Access, dando origem a uma tabela cuja estrutura é composta por todos os campos dos cabeçalhos IP e TCP. A seguir é descrito o que foi verificado no esquema proposto para realização da análise com estado, o ESTCON.

3.5.1. Classificação dos pacotes por Conexão

Como a análise com estado tem como foco central a conexão TCP, a primeira fase que compõe o ESTCON consiste na classificação dos pacotes de acordo com a conexão à qual eles pertencem (ou deveriam pertencer).

Essa classificação é feita através do “par socket”, que consiste nos endereços de origem e destino do cabeçalho IP, e nas portas de origem e destino do cabeçalho TCP, o que identifica uma conexão de forma única na rede.

Concluída a classificação dos pacotes por conexão, que é realizada através de uma consulta no banco de dados agrupando-se os pacotes pelo respectivo “par socket”, é iniciado um processo de verificação de quatro itens para cada conjunto de pacotes identificado como sendo uma conexão, os quais são descritos na seção seguinte.

3.5.2. Itens Verificados no ESTCON

A base do ESTCON consiste na verificação de quatro itens referentes aos comportamentos mais genéricos da conexão TCP/IP, definidos em [Northcutt, 2000], os quais são identificados na Tabela 2.

Tabela 2. Tabela-Verdade dos Quatro Itens verificados no ESTCON

	Item 1	Item 2	Item 3	Item 4		
Cód.	**Estab. de Conexão**	**Transf. de Dados**	**Flag FIN**	**Flag RST**	**SIGNIFICADO**	**STATUS**
1	V	V	V	V	Três fases completas	NORMAL
2	V	V	V	F	Três fases completas	NORMAL
3	V	V	F	V	Interrupção abrupta (possível falha de operação)	NORMAL
4	V	V	F	F	Transferência de dados não terminada (Cold End)	NORMAL
5	V	F	V	V	Inicia e finaliza conexão, sem transferência de dados	ANORMAL
6	V	F	V	F	Inicia e finaliza conexão, sem transferência de dados	ANORMAL
7	V	F	F	V	Interrupção abrupta antes de iniciar transferência de dados (possível falha de operação)	NORMAL
8	V	F	F	F	Conexão ainda não utilizada	VERIFICAR
9	F	V	V	V	Não aparece estabelecimento da conexão (Cold Start)	NORMAL
10	F	V	V	F	Não aparece estabelecimento da conexão (Cold Start)	NORMAL
11	F	V	F	V	Cold Start e término abrupto	NORMAL
12	F	V	F	F	Cold Start e Cold End	NORMAL
13	F	F	V	V	Só aparece finalização da conexão	VERIFICAR
14	F	F	V	F	Só aparece finalização da conexão	VERIFICAR
15	F	F	F	V	Resposta a pedido não aceito	VERIFICAR
16	F	F	F	F	Desconhecido	SUSPEITO

No item 1 é verificado se ocorreu o estabelecimento de conexão através do processo de *three-way handshake*. No item 2 é verificado se houve transferência de dados em alguma direção com a respectiva confirmação de recebimento, dentro da conexão.

No item 3 é verificado se houve solicitação de finalização de conexão através do uso do flag FIN. Finalmente, no item 4, é verificado se o flag RST foi utilizado em algum momento dentro da conexão.

Caso o resultado da verificação do item seja positivo, o mesmo recebe valor “V” (verdadeiro); caso contrário, recebe valor “F” (falso).

Assim, de posse dos valores obtidos nos quatro itens, pode-se identificar o que ocorreu dentro de cada conexão de acordo com a Tabela 2. Tal tabela apresenta as 16 (dezesseis) combinações possíveis referentes aos quatro itens examinados, correspondendo, dessa forma, a uma tabela-verdade dos itens do ESTCON. O conteúdo da coluna “Significado” é inferido a partir da combinação dos valores dos quatros itens,

cujo objetivo maior é descrever de forma resumida o que aconteceu dentro do conjunto de pacotes analisado.

O termo "Cold Start", utilizado na Tabela 2, indica a situação em que uma conexão foi estabelecida antes do tráfego da rede começar a ser monitorado [Handley e Paxson, 2001]. Analogamente, o termo "Cold End" será utilizado com o significado de que a conexão ainda não havia sido terminada no momento em que o arquivo de log foi finalizado.

A seção seguinte trata do valor de status atribuído com base no resultado da verificação dos quatro itens da Tabela 2.

3.5.3. Definição de Status proposta no ESTCON

O valor de "STATUS", definido na Tabela 2, baseia-se no conhecimento do comportamento padrão dos protocolos IP e TCP, bem como em algumas técnicas de ataque que se manifestam mediante a violação de tais padrões.

O status "NORMAL" significa que o conjunto de pacotes analisado apresenta um comportamento aceitável ou esperado pelos protocolos TCP/IP. O status "SUSPEITO" significa que o conjunto de pacotes analisado apresenta um comportamento que representa uma possível forma de ataque (por exemplo, uma técnica de scan). Já o status "ANORMAL" significa que o conjunto de pacotes analisado não apresenta um comportamento padrão, não chegando, porém, a ser considerado suspeito.

Finalmente, o status "VERIFICAR" significa que somente com base nos valores dos quatro itens da tabela-verdade não é possível identificar o real status do conjunto de pacotes analisado, sendo necessário, portanto, verificar algumas questões relacionadas a seguir.

No caso do código 8 da Tabela 2, é verificado se o último pacote, presente no conjunto sob análise, localiza-se no final do log. Em caso positivo, o status da conexão é considerado "NORMAL", assumindo-se que o encerramento do log ocorreu antes que as fases posteriores (transferência e finalização) pudessem ser registradas; caso contrário, o status é dito "ANORMAL". Para efeitos do ESTCON, a verificação do posicionamento do pacote dentro do log dá-se em razão do total **N** de pacotes armazenado no mesmo, dividindo-se o log em três partes iguais (início, meio e fim), cada uma contendo (**N***33,33%) pacotes.

No caso dos códigos 13 ou 14, é verificado se o primeiro pacote, presente no conjunto sob análise, localiza-se no início do log. Em caso positivo, o status da conexão é considerado "NORMAL", assumindo-se que ocorreu "Cold Start"; caso contrário, o status é dito "ANORMAL".

No caso do código 15, que significa que dentre os quatro itens só foi constatado o uso do flag RST, primeiramente é verificado quais pacotes, dentro do conjunto sob análise, deram origem aos pacotes RST. Com base nestes pacotes, é verificado se o <End_Origem> presente nos mesmos se repete em outras "conexões" também classificadas com código 15. Caso o número de repetições ultrapasse um limiar pré-definido (valor *default* 10), o status da conexão é considerado "SUSPEITO", pois isto significa que um mesmo <End_Origem> enviou vários pacotes para diversos destinos, caracterizando um *scan* a partir do <End_Origem> em questão.

Caso um *scan* não seja configurado, é feita uma verificação da quantidade de pacotes que originaram os pacotes RST, por grupos de pacotes. Caso essa quantidade seja superior a um limiar (valor default 300), o status da conexão é considerado "SUSPEITO", configurando-se um ataque de negação de serviço (DoS) através da inundação (*flooding*) da porta destino com inúmeras requisições [Northcutt, 2000].

Caso um *flooding* não seja configurado, é verificado qual flag foi utilizado para gerar o pacote RST. Se flag=SYN, o status é considerado "NORMAL", assumindo-se que se trata de uma tentativa de conexão não aceita. Se flag=ACK, é verificado se o pacote localiza-se no início do log, caso positivo, o status é considerado "NORMAL", assumindo-se que ocorreu "Cold Start" com término abrupto, caso negativo, o status é dito "SUSPEITO", assumindo-se que houve tentativa de reconhecimento através do uso do flag ACK sem uma conexão pré-estabelecida. Por fim, se flag=FIN ou qualquer outro valor desconhecido, o status é considerado "SUSPEITO" pelo mesmo motivo de provável tentativa de reconhecimento.

4. Resultados Obtidos

Foram realizados vários testes com o RECAB e o ESTCON envolvendo tráfego real de rede e pacotes montados contendo anomalias e simulando algumas formas de ataques. Para efeitos de comparação, os mesmos testes foram feitos com o Snort, que é um sistema de detecção de intrusão de domínio público baseado em assinaturas de ataques.

Todos os testes foram realizados em ambiente Windows 2000, com interface de rede Ethernet de 100 Mbits/sec. Foi utilizado Snort para Windows versão 1.9.1, disponível no endereço eletrônico <http://www.silicondefense.com/support/window>. A seguir são apresentados os testes realizados.

4.1. Testes com o RECAB

Primeiramente, foram feitos testes com 100.000 pacotes coletados do tráfego real de uma rede TCP/IP. Como resultado, o RECAB detectou alguns pacotes contendo violações de protocolo, onde a maioria (202 pacotes) apresentava algum valor no campo <Pont_Urg> (*urgent pointer*) do cabeçalho TCP, apesar do flag URG estar zerado. Já o Snort, só detectou alguns pacotes RST considerados "clandestinos" (66 pacotes).

Entretanto, apesar dos alertas gerados tanto pelo RECAB quanto pelo Snort, indicando, respectivamente, a existência de cabeçalhos incorretos e de pacotes RSTs indevidos, não ficou constatada nenhuma má intenção, não sendo, portanto, identificada qualquer tentativa de ataque.

Portanto, a partir desses resultados, percebe-se que, para efeito de realização de testes para detectar violações de protocolos e possíveis ataques, o ideal seria que a análise fosse realizada sobre um conjunto de pacotes montados especificamente para testar as regras do protótipo e simular algumas formas de ataque.

Dessa forma, foram gerados vários pacotes contendo diversos tipos de anomalias para serem submetidos à análise. Os resultados mais relevantes obtidos com a análise destes pacotes, pelo RECAB e pelo Snort, são apresentados na Tabela 3.

Tabela 3. Comparação dos Resultados do RECAB e do Snort referente à Análise de Pacotes Gerados com Anomalias.

Seq.	Características do pacote testado	Comportamento do RECAB	Comportamento do Snort
01	Bit ACK = 0 e Num_ACK > 0	Bit ACK e Num_ACK Inválidos	<sem-alerta>
02	Bit URG = 1 e <Pont_Urg> = 0	Bit URG e Pont_URG Inválidos	<sem-alerta>
03	Flag URG = 1 em pacote sem dados	Uso de flag URG sem envio de dados	<sem-alerta>
04	Flags SYN-FIN setados	FlagsTCP: Combinação Inválida	Scan SYN-FIN
05	Flag ACK=1 e Num_ACK=0	Bit ACK e Num_ACK Inválidos	Scan nmap
06	Flags SYN-FIN-ACK setados e Num_ACK=0	FlagsTCP: Combinação Inválida Bit ACK e Num_ACK Inválidos	STEALTH ACTIVITY (unknown)
07	<End_Origem> = <End_Destino>	Endereços de Origem e Destino Iguais	BAD TRAFFIC same SRC/DST
08	<End_Origem> = <End_Destino> e Flag ACK=1 e Num_ACK=0	Endereços de Origem e Destino Iguais Bit ACK e Num_ACK Inválidos	Scan nmap
09	Bit DF=1 e Bit MF=1 do campo <FlagsOffset> e Flags SYN-FIN setados	FlagsOffset: Bit DF=1 e Bit MF=1 FlagsTCP: Combinação Inválida	BAD TRAFFIC bad frag bits
10	Flag SYN=1 em pacote com dados	Dados enviados em pacote SYN	BAD TRAFFIC data in TCP SYN packet
11	Flags SYN-FIN setados em pacote com dados	FlagsTCP: Combinação Inválida Dados enviados em pacote SYN	Scan SYN-FIN
12	Flags SYN-ACK setados em pacote com dados	Dados enviados em pacote SYN	<sem-alerta>

O RECAB mostrou-se hábil em identificar violações de regras de protocolo, com a característica de que, a cada pacote, todas as regras eram verificadas. Esta característica permitiu a detecção de ataques que combinavam mais de uma assinatura, referentes à violação de protocolo, em um único pacote, fato este que não foi observado na análise realizada pelo Snort, conforme mostra o item 06 da Tabela 3. Além disso, para algumas violações de protocolo o Snort sequer gerou alerta.

Outra vantagem observada no RECAB em relação ao Snort, diz respeito ao mecanismo de detecção do próprio Snort, que finaliza no momento em que a primeira assinatura é detectada. Assim, se um pacote é composto por mais de uma assinatura de ataque que utiliza campos diferentes dos cabeçalhos (itens 08, 09 e 11), somente a primeira é detectada pelo Snort, ao contrário do RECAB que faz uma análise completa dos cabeçalhos.

Entretanto, deve-se observar que o protótipo mostrou um baixo desempenho no tratamento de grandes quantidades de pacotes. Assim, há de se considerar, para uma aplicação prática, uma customização da forma de programação e, até mesmo, a utilização de outra linguagem.

4.2. Assinaturas do Snort *versus* Violação dos Protocolos IP e TCP

Além dos testes com o RECAB, foi feito um levantamento das assinaturas presentes no Snort, sendo verificado que poucas se referem à violação de protocolo, onde a maioria realiza buscas por strings suspeitas na parte de dados dos pacotes.

Entretanto, mesmo que para uma minoria, foi verificado que algumas regras tratadas no RECAB poderiam representar um dado conjunto de assinaturas do Snort, chegando-se numa proporção de 1:12 (uma regra para doze assinaturas). Assim, através

de uma pré-análise, utilizando-se o RECAB, seria possível diminuir o número de assinaturas em um SDI, o que, dependendo da proporção, incrementaria a eficiência do mecanismo de detecção.

4.3. Testes com o ESTCON

Primeiramente, foram feitos testes com os mesmos 100.000 pacotes utilizados no RECAB, onde, após a classificação dos pacotes por "par socket", foi identificada uma situação com status "SUSPEITO", referente ao código 15 da Tabela 2.

Verificou-se que a partir de um mesmo endereço de origem (host-a) foram enviados mais de 600 pacotes SYN, a 66 endereços de destino diferentes em portas diversas. Entretanto, nenhuma das solicitações de conexão foi concluída, sendo enviado como resposta, por parte do endereço de destino, na grande maioria das vezes, pacotes RST. Tal situação foi caracterizada pelo ESTCON como um *scan* partindo do host-a, onde o intuito seria identificar quais portas estariam ativas ou não em diversos hosts.

Entretanto, ao contactar o administrador da rede, verificou-se que o host-a correspondia a um servidor NAT (*Network Address Translation*) cujo endereço IP é utilizado por todos os hosts da rede interna para acesso à Internet. Assim, concluiu-se que não se tratava de um *scan* partindo de uma mesma origem, mas sim de vários hosts internos tentando acessar diversos destinos, gerando, portanto, um FALSO-POSITIVO.

Para testar devidamente o ESTCON, foram geradas algumas seqüências de pacotes simulando ataques, utilizando-se as ferramentas *NMapWin* (www.insecure.org) e *Engage Packet builder* (www.engagesecurity.com). Os resultados obtidos no ESTCON e no Snort são apresentados na Tabela 4.

Tabela 4. Resultados obtidos no ESTCON e no Snort referente a Pacotes Montados Simulando Ataques.

Ataque	ESTCON	Snort
Scan SYN Stealth (via *NMapWin*)	Scan a partir da Origem: "host-y" - Flags **SYN**, ACK, NULO, URG-PSH-FIN (Tentativas: 1616)	NMAP FINGERPRINT (stateful) STEALTH ACTIVITY (XMAS scan) STEALTH ACTIVITY (NULL scan)
Null Scan (via *NMapWin*)	Scan a partir da Origem: "host-y" - Flags **NULO**, ACK, URG-PSH-FIN (Tentativas: 1881)	NMAP FINGERPRINT (stateful) STEALTH ACTIVITY (XMAS scan) Gerou mais de mil alertas do tipo: STEALTH ACTIVITY (NULL scan)
ACK Scan (via *NMapWin*)	Scan a partir da Origem: "host-y"- Flags **ACK**, URG-PSH-FIN (Tentativas: 1876)	Gerou três alertas do tipo: STEALTH ACTIVITY (XMAS scan)
SYN-flood (via *Engage Packet builder*)	Denial of Service (flooding) no "host-x:porta-x" a partir da Origem: "host-y" - Flags **SYN** (Tentativas: 800)	<sem-alerta>

Ao utilizar a ferramenta *NMapWin* observou-se que ela sempre enviava em seus "*scans*" algum pacote contendo um tipo de anomalia (flag NULO ou flag ACK com <Num_Ack>=0 ou flag URG-PSH-FIN), não se restringindo a enviar pacotes com flags setados de acordo com a respectiva denominação do scan. Esses "scans" eram enviados a partir de uma origem para diversas portas em único host com o objetivo de identificar o sistema operacional rodando no host destino (*fingerprinting*). Já o ataque SYN-flood consistiu do envio de 800 pacotes SYN para uma mesma porta em um único host, com objetivo de gerar um Denial of Service (DoS) na porta de destino.

Assim, verifica-se que o Snort, apesar de ter gerado alertas para os três scans, gerou-os em decorrência dos pacotes anômalos enviados pelo *NMapWin*, os quais, na realidade, não deveriam fazer parte destes scans. Portanto, à exceção do Scan Null, conclui-se que o Snort não detectou os demais scans, nem tampouco o SYN-flood.

No ESTCON, foi atribuído status "SUSPEITO" para as quatro situações mostradas na Tabela 4. Nos três primeiros casos, foi detectado um scan a partir do "host-y", identificando-se os flags utilizados na varredura e o número de pacotes enviados, onde o flag em negrito refere-se àquele que apareceu em maior quantidade. No caso do SYN-flood, identificou-se o destino e a origem da inundação (*flooding*), o flag utilizado e o número de pacotes enviados.

4.4. Comparação do ESTCON com um Sistema de Reconstrução de Sessões TCP/IP denominado RECON

O RECON [Chaves, 2002] é um sistema que permite reconstruir e rastrear o estado das sessões TCP/IP, utilizando o cabeçalho dos pacotes, e que implementa algumas rotinas para realizar aplicações em detecção de intrusão. Uma dessas rotinas tem o objetivo de realizar Detecção de *Host Scan* com base nas sessões reconstruídas pelo RECON, a qual foi comparada com a detecção de *scan* feita pelo ESTCON.

O RECON faz a verificação da quantidade de hosts acessados a partir de um mesmo IP utilizando uma lista de sessões TCP criada quando da análise do tráfego, onde a porta de destino tem que ser igual para todos os hosts acessados. Já no ESTCON, a mesma verificação de quantidade de hosts é feita somente onde não há sessão TCP estabelecida (código 15 da Tabela 2), não sendo necessário que a porta de destino seja a mesma, bastando somente que a "conexão" (par socket) seja diferente.

Assim, no caso de uma varredura em único host em todas as portas (1 a 65535), para saber quais serviços o mesmo oferece, o RECON não a detectaria. No ESTCON, mesmo o destino sendo um único host, como as portas variam, os pacotes são agrupados em "conexões" diferentes. Dessa forma, o ESTCON identifica várias solicitações de conexões partindo de um mesmo host com destinos diferentes, sem que nenhuma seja estabelecida, o que caracteriza o *scan*.

Com relação a desvantagens, pode-se citar a questão de que o RECON também analisa sessões TCP estabelecidas na detecção de scan, o que não ocorre no ESTCON. Um exemplo de dados reais, descrito em [Chaves, 2002], apresenta uma atividade onde um mesmo host-x iniciou sessões com vários hosts na porta 25/tcp, a qual ocorreu durante 40 horas consecutivas, com picos de até 807 acessos a hosts distintos. Tal situação, que não seria detectada pelo ESTCON, foi detectada pelo RECON, sendo constatado que o host-x estava sendo explorado e utilizado para gerar SPAM.

5. Conclusão

A análise de pacotes proposta permitiu que fossem criados uma definição formal das especificações dos protocolos IP e TCP e um esquema de se fazer correlação de pacotes, que se mostraram eficientes não só para realizar detecção de alguns tipos de ataques como também para estudar o comportamento do tráfego de redes TCP/IP.

A maior vantagem observada em verificar a violação de protocolos, consiste no fato de que as possibilidades de utilização de anomalias para geração de ataques seriam

esgotadas. Assim, variações de ataques e ataques desconhecidos, que envolvam violação de protocolo, poderiam ser detectados sem a pré-existência de uma assinatura.

Como trabalhos futuros, as regras do RECAB poderiam ser implementadas como um *plugin* do Snort [Roesch e Green, 2003]. Com relação ao ESTCON, poderia ser feito um estudo da taxa de falso-negativos ao se aplicar as inferências apresentadas na Tabela 2. Além disso, para ampliar a detecção de ataques, uma análise de anomalias referente a freqüência e tipos de acesso, envolvendo um processo de aprendizado do perfil de tráfego, poderia ser realizada após as verificações do ESTCON. Uma última sugestão consiste em aumentar o escopo da análise de pacotes, englobando outros protocolos, tais como ICMP e UDP.

Referências

Allen, J. et al. *State of the Practice of Intrusion Detection Technologies* [online]. Pittsburgh: Carnegie Mellon University, 2000. <http://www.sei.cmu.edu/publications /documents/99.reports/99tr028/99tr028abstract.html>

Amoroso, Edward, *Intrusion Detection, An Introduction to Internet Surveillance, Correlation,TraceBack,Traps,and Response*. New Jersey:Intrudion.Net Books, 1999.

Casanova, Marco A. et al. *Programação em Lógica e a Linguagem Prolog*. São Paulo: Editora Edgard Blücher Ltda, 1987.

Chaves, Marcelo H. P. C. *Análise de Estado de Tráfego de Redes TCP/IP para Aplicação em Detecção de Intrusão*. São José dos Campos: Instituto Nacional de Pesquisas Espaciais, 2002. 172 p. (INPE-9625-TDI/845).

Handley, Mark and Paxson, Vern. *Network Intrusion Detection: Evasion, Traffic Normalization, and End-to-end Protocol Semantics* [online]. Proc. USENIX Security Symposium 2001. <http://www.icir.org/vern/papers.html>.

Krügel, Christopher et al. *Service Specific Anomaly Detection for Network Instrusion Detection* [online]. <http://www.informatik.uni-trier.de/~ley/db/conf/sac/ sac2002.html#KrugelTK02> (2002).

Northcutt, Stephen et al. *Intrusion Signatures and Analysist´s Handbook*. Indiana: New Riders Publishing, 2001.

Northcutt, Stephen et al. *Network Intrusion Detection – An Analyst´s Handbook*. Indiana: New Riders Publishing, 2000.

Roesch, M. and Green, C. *Snort Users Manual. Release 2.0.0*. [online]. <http://www.snort.org/docs/SnortUsersManual.pdf>. Abril, 2003

Sager, Naomi. *Natural Language Information Processing*. Massachusetts: Addison-Wesley Publishing Company, 1981.