Noname manuscript No.
(will be inserted by the editor)

SISTEMA PARA SUPORTE À DETECÇÃO DE ATAQUES VINDOS DA INTERNET

Lucas Vasconcelos Santana · Milton Shimabukuro

Received: 10/26/2012 / Accepted: 10/26/2012

Resumo Desenvolvimento de IDS (Intrusion Detection System) do tipo Network-based. Capaz de detectar por assinatura ameaças do tipo PortScan e SYN Flood. Geração de relatórios em texto e com visualização gráfica para o adminitrador. Alerta o administrador sobre os eventos ocorridos.

Keywords IDS · Segurança · Ataques · Ameaças

Lucas Vasconcelos Santana
E-mail: luksvs@gmail.com
Milton Shimabukuro
E-mail: miltonhs@fct.unesp.br

1 Introdução

Texto da introdução.

2 Artigos relacionados

2.1 IDS

A proposta do projeto é o desenvolvimento de um IDS (Intrusion Detection System ou Sistema de detecção de intrusos). Um IDS deve ser capaz de detectar ataques e comportamentos estranhos no ambiente em que ele se encontra, registrar os eventos e alertar seu administrador. Um IDS pode possuir diversas características e modos de funcionamento, como pode ser visto na Fig. 1.

Entretanto, nem todas essas características serão explicadas nesse documento. Aqui serão detalhadas as que possuem importância no assunto abordado e no desenvolvimento do projeto em si. Algumas características incomuns e pouco utilizadas não fazem parte do escopo do documento.

Para executar sua principal função, que é a detecção de ataques, um IDS pode utilizar dois métodos de detecção: baseado em assinatura e baseado em anomalias. O sistema pode fazer uso de apenas um desses métodos, que é o cenário mais comum, ou ser híbrido, utilizando os dois métodos.

2.1.1 Detecção baseada em assinatura

Uma assinatura é um comportamento padrão de um ataque conhecido. O IDS que utilizar esse método para suas detecções irá possuir uma base com as assinaturas de cada tipo de ataque conhecido ou que ele se propõe a detectar. A partir dessas assinaturas, o sistema será capaz de perceber as tentativas de ataques que são efetuadas.

Um exemplo de assinatura é a utilizada na detecção de um PortScan [7], onde o IDS irá verficar se um determinado host está executando um PortScan em seu ambiente, analisando as portas de destino dos pacotes. Caso haja incrementos ou decrementos de uma unidade nas portas de destino, isso pode ser considerado um PortScan.

Esse tipo de método consegue bastante sucesso na detecção de ameaças conhecidas, mas não tem êxito na detecção de ataques desconhecidos ou que possuem procedimentos esparsos.

2.1.2 Detecção baseada em anomalias

Utilizando esse método de detecção, o IDS irá analisar seu ambiente durante um determinado período de tempo. Essa análise é chamada de período de treinamento. Após essa análise, serão gerados perfis baseados nos comportamentos que foram analisados.

Com os perfis criados, o IDS vai comparar o comportamento encontrado em seu ambiente com os perfis criados no período de treinamento. Caso algum evento anômalo seja percebido, o IDS vai alertar o administrador.

Um exemplo de detecção baseada em anomalia, é um IDS que gera um perfil que mostra que o uso médio da banda de internet de uma empresa em um certo horário é de 13% [10], se em uma semana o IDS perceber que o uso médio da banda de internet subiu para 47%, o administrador será alertado.

IDSs que usam esse método apresentam bons resultados na detecção de ameaças não conhecidas ou criadas recentemente. Porém, esse tipo de detecção causa muitos falsos-positivos. Podemos usar como exemplo um evento que acontece a cada 30 dias, que é um envio ou recebimento de grande quantidade de dados [10]. As chances desse evento não ser analisado durante o período de treinamento são grandes, e no momento que ele ocorrer o IDS irá identificá-lo como uma anomalia, caracterizando o evento como um ataque. O aumento do período de treinamento seria uma solução óbvia para esse problema, mas na prática é inviável utilizar um longo período de treinamento.

Do ponto de vista do desenvolvimento, esse método também oferece dificuldades, visto que a análise que precisa ser executada durante o período de treinamento pode ser complexa computacionalmente.

2.2 Tipos de IDS

Scarfone & Mell, em seu Guia para Sistemas de Detecção e Prevenção de Intrusos [10], definem os tipos de IDS da seguite maneira:

- **Network-based:** Aonitora o tráfego de uma determinada rede ou interface de rede e analisa os pacotes e os protocolos de aplicação para identificar atividade suspeita. Um IDS desse tipo pode detectar muitos tipos de eventos de interesse para o administrador. Normalmente ele se encontra nas bordas das redes, junto com firewalls e roteadores, servidores de VPN, servidores com acesso remoto e redes sem fio.
- **Wireless:** Analisa todo o tráfego de uma rede sem fio, verficiando seus protocolos e procurando atividade suspeita nos mesmos. Esse tipo de IDS não

Figura 1 Características de um IDS [12]

consegue identificar ameaças nas camadas de Aplicação e Transporte, como o tipo anterior. Geralmente é implantado para monitorar redes sem fio de empresas e em redes que possuem risco de acesso não autorizado.
- **Network Behavior Analysis (NBA ou Análise de Comportamento da Rede):** examina o tráfego de uma rede a procura de ameaças que podem gerar anomalias de fluxo no tráfego, como ataques DDoS, malwares e violações na política de segurança do ambiente. IDS desse tipo costumam ser implantados dentro das redes de organizações, buscando anomalias na comunicação de dentro para fora, como por exemplo os acessos à Internet e à rede de uma filial.
- **Host-based:** monitora as caracterísicas de um único host e os eventos relacionados a este por atividades suspeitas. Esse tipo de IDS faz análises do tráfego da rede (apenas desse host), logs do sistema, processos que são executados, atividades das aplicações, acessos aos arquivos e mudanças nas configurações do sistema. Eles são implantados em hosts que exigem um certo nível de segurança, como um servidor com acesso público ou que possua dados importantes.

2.3 Funções básicas de um IDS

Não existem regras para o desenvolvimento de um IDS, mas é desejável que ele possua algumas funções básicas. São elas:

- Capacidade de analisar os eventos ocorridos em seu ambiente de alguma maneira, seja capturando pacotes ou analisando registros e logs do sistema;
- Capacidade de detectar ameaças e ataques em seu ambiente;
- No caso de haver alguma detecção, o IDS deve registrar as informações pertinentes relacionadas ao evento;
- Ele também deve alertar o administrador sobre as ameaças que foram detectadas;

2.4 Desafios no desenvolvimento de um IDS

2.4.1 Criptografia nos dados

Silveira, em seu artigo "Desafio para os Sistemas de Detecção de Intrusos" [11] diz:

Figura 2 SSL nas camadas TCP/IP [11]

Figura 3 Transport Mode [11]

"As ferramentas de IDS baseadas em rede fazem suas monitorações nos cabeçalhos dos pacotes e também em seu campo de dados, possibilitando a verificação de ataques no nível de aplicação (para pacotes TCP e UDP).

Entretanto, a necessidade de sigilo e privacidade nas transações pela Web ou em redes privadas torna o uso de sistemas de criptografia cada vez mais comuns e necessários.

A criptografia, como elemento de segurança do tráfego de informações, acaba por prejudicar outros elementos como os sistemas de detecção de intrusos (IDS). Uma vez que, em algumas tecnologias, o campo de dados é criptografado, em outras, o pacote inteiro o é (cabeçalhos e dados). Neste ambiente, uma ferramenta de IDS não será efetiva, pois os dados de um ataque podem ser encobertos pela criptografia existente no mesmo."

Duas formas de criptografias bem conhecidas são o SSL e o IPSec, que são geralmente usados no protocolo HTTP e na criptografia de VPNs, respectivamente.

2.4.2 SSL

O SSL (Fig. 2) acrescenta uma camada entre o protocolo da camada de Aplicação e o protocolo da camada de Transporte que esteja sendo utilizado (transporte confiável), como por exemplo, o protocolo TCP.

O SSL criptografa o pacote TCP antes do envio do mesmo. Dessa maneira, um IDS não consegue analisar o conteúdo do pacote e detectar um possível ataque.

Nesse cenário, é fácil ver que um pacote que utilize o SSL para chegar até a camada de Aplicação do seu destino, pode ser usado para um ataque efetivo.

2.4.3 IPSec

O IPSec é um técnica que funciona em cima do protocolo IP, que consegue criptografar e assinar os pacotes IP. Ela é amplamente utilizada para a garantia de segurança em soluções de VPN (Virtual Private Network).

O IPSec pode funcionar de dois modos: o de transporte (transport mode) e o túnel (tunnel mode).

O IPSec em Modo Transporte (Fig. 3) se assemelha ao SSL, onde ele faz a criptografia de apenas parte do pacote IP. Portanto, ainda é possível o acesso ao cabeçalho do pacote IP.

Figura 4 Tunnel Mode [11]

No Modo Túnel (Fig. 4), o IPSec criptografa todo o pacote IP, impossibilitando a análise até do cabeçalho do pacote.

Enfim, os IDS também terão problemas ao tentar a análise de pacotes em que o IPSec está sendo utilizado, pois não conseguirão verificar o conteúdo do pacote e até mesmo o próprio cabeçalho do pacote.

2.4.4 Redes de alta velocidade

Cada vez mais a largura de banda utilizada para Internet e para as redes internas estão aumentando. Quandidades antes inimagináveis de tráfego passam normalmente por interfaces de redes dos servidores.

Esse é um outro problema encontrado pelos desenvolvedores de IDS. Os algoritmos de detecção já são consideravelmente complexos, e utilizá-los em uma rede com tráfego excessivo pode ser uma tarefa complicada computacionalmente.

Uma saída para esse problema, é a implantação de IDS do tipo Host-based, que são somente responsáveis pelo host em que eles se encontram.

3 Proposta do projeto

O projeto propõe o desenvolvimento de um IDS do tipo Network-based cujo método de detecção é baseado em assinaturas. Ele possuirá todas as funções básicas supracitadas e a princípio possuirá as assinaturas para detecção dos ataques do tipo PortScan e DdoS (do tipo SYN Flood).

Um IDS com essas características é comumente chamaod de NIDS (Network Intrusion Detection System).

O IDS terá que analisar os dados em tempo real, por meio de captura de pacotes de uma determinada interface ou endereço IP.

Também é desejável para o sistema a geração de relatórios que podem ser visualizados com auxílio de ferramentas de visualização voltadas para segurança de redes.

4 Fundamentação teórica

4.1 GNU/Linux

O sistema operacional que será utilizado no trabalho é o Linux, que é um sistema operacional desenvolvido por Linus Torvalds baseado no kernel Minix. O nome Linux originalmente era dado apenas para o kernel do sistema, mas atualmente o nome é atribuído para o sistema como um todo. O Linux é um sistema de código livre (opensource) e se encontra disponível sob a licença GPL versão 2.

O Sistema Operacional Linux é amplamente utilizado em servidores devido a sua segurança e estabilidade, e a comunidade que mantém o Linux e suas ferramentas é muito ativa e atualizações de segurança surgem constantemente. Uma desvantagem para as empresas que escolhem o Linux é a falta de profissionais capacitados que conseguem usufruir de toda capacidade e funcionalidades que o sistema pode oferecer.

Uma pesquisa feita pela W3Techs [13] indica que 93,3% de servidores Web rodam Linux/Unix.

Para o desenvolvimento do projeto, será necessário um conhecimento intermediárioavançado sobre o Linux e suas ferramentas. Pois um servidor Firewall usando Linux deverá ser desenvolvido para simulações do sistema. Também serão feitas tentativas de invasões simuladas a partir de outro computador com o sistema Linux instalado.

4.2 Linguagem C

A linguagem C é considerada uma linguagem de baixo-nível, ideal para desenvolver sistemas que fazem comunicação com hardware e ferramentas de rede, [2] oferece material completo para o estudo dessa linguagem.

A linguagem C, criada em 1972 no AT&T Bell Labs, e usada para desenvolver o sistema operacional Unix, sendo a linguagem preferida entre os desenvolvedores desse tipo de aplicação, será utilizada para o desenvolvimento do IDS.

Conhecimentos avançados em desenvolvimento com a linguagem C serão exigidos, tal como o uso de sockets para desenvolvimento de aplicações que utilizam os protocolos de rede e manipulação de arquivos.

4.3 Modelo TCP/IP

Conhecer o modelo TCP/IP é importante no desenvolvimento de um NIDS. Esse modelo é o atual modelo da internet, pois oferece serviços robustos e envio e recebimento de dados de forma correta e sem erros.

O modelo TCP/IP é composto por quatro camadas, onde cada qual possui seus próprios protocolos com funções específicas.

Camada	Descrição
4- Aplicação	Comunicação entre processos, normalmente os procotolos em que o usuário possui contato, como HTTP, HTTPS, FTP, SMTP, etc.
3 - Transporte	Oferece serviço para transporte de dados com confiabilidade e integridade. Os protocolos mais utilizados nessa camada são o TCP e o UDP.
2 - Internet	É aqui que se encontra o protocolo IP, que faz o roteamento dos pacotes entre as diversas redes encontradas na Internet. Outro protocolo conhecido dessa camada é o ICMP.
1 - Rede	Em comparação com o modelo OSI/ISO, essa camada faz correspondência com as camadas de Enlace e Fisica. Ela é responsável pelo transporte dos dados de um ponto ao próximo ponto. O protocolo predominante nessa camanda é o Ethernet.

Para o desenvolvimento de um NIDS com as características desejadas, são interessantes as camadas de Transporte e de Internet. Cada uma delas pode oferecer dados contidos nos cabeçalhos de seus protocolos que são importantes para as detecções baseadas em assinatura que o sistema se propõe a fazer.

4.4 Ferramentas de visualização

Como foi dito anteriormente, é desejável que o sistema consiga gerar relatórios que possam ser visualizados com a ajuda de ferramentas. Será selecionada uma das ferramentas que compõe o projeto Davix [14], que possui várias aplicações para geração de gráficos baseados em informações existentes. Tais visualizações são voltadas para a área de segurança.

Figura 5 Ataque do tipo SYN Flood [19]

4.5 Estudo sobre os ataques

O IDS a desenvolvido nesse projeto fará suas detecções baseado em assinaturas. Portanto, o desenvolvedor deve conhecer os ataques que o sistema se propõe a detectar a fim de criar os algoritmos necessários para as assinaturas de cada um dos ataques.

PortScan: utilizando um programa, o hacker faz uma varredura em todas as portas (1 a 65535) de um determinado host. Utilizando o TCP Connect ele verificar se cada porta está aberta. Caso ele encontre portas abertas na vítima, ele pode tentar fazer uma invasão por ela.

DDoS (SYN Flood - Fig. 5): Existem diversas maneiras de se efetuar um ataque do tipo DDoS (Negação de Serviço), e uma delas é utilizando o método SYN Flood. O IDS deve ser capaz de detectar um ataque desse tipo, onde o hacker dispara vários pacotes TCP com a flag SYN, impossibilitando que a vítima processe todos os pacotes. Dessa maneira, a vítima fica sobrecarregada e não consegue processar novas requisições, ocasionando a parada do serviço que ela executa.

5 Arquitetura e estrutura do sistema

5.1 Decisões técnicas

Foram feitas diversas pesquisas e análises em artigos e soluções existentes de IDS com as características desejadas para a escolha das tecnologias e ferramentas que foram usadas e que são citadas na seção 4.

As próximas subseções irão mostrar os passos do desenvolvimento em ordem cronológico, mostrando os problemas encontrados e as possíveis saídas.

5.2 Captura de pacotes

Nas seções anteriores são encontradas os motivos das escolhas do Sistema Operacional e da linguagem de programação que foram utilizadas no projeto. Porém, o primeiro problema encontrado foi como executar a captura de pacotes em tempo real, que é uma proposta do projeto.

A melhor opção encontrada foi a biblioteca para C/C++ LibPcap [20], que é uma biblioteca feita para a captura de pacotes de uma rede ou interface. Apesar dela trabalhar num nível baixo, aumentando a complexidade dos códigos e do aprendizado, ela oferece ótimo desempenho e ótima documentação. Um analisador de

pacotes bem conhecido que utiliza a LibPcap é o Tcpdump [20]. Outra boa característica da LibPcap é o fato dela ser opensource (código livre).

A partir de estudos e da leitura das documentações encontradas na Internet e oferecidas pelos próprios mantenedores do projeto, foi possível o trabalho com a LibPcap no desenvolvimento do projeto.

Seguem as características resumidas de cada uma das funções que a LibPcap ofereceu para o desenvolvimento do projeto:

- pcap_lookupnet(): Permite a obtenção de informações de uma determinada interface, tal como a máscara e a rede que ela se encontra.
- pcap_open_live(): Permite que seja possível a captura de pacotes em uma interface.
- pcap_setfiler(): Permite que sejam incluídos filtros nas capturas dos pacotes, essa função deve funcionar em conjunto com a função pcap_compile().
- pcap_loop(): Faz com que a cada pacote que seja capturado pela interface escolhida uma função do tipo callback seja chamada. É nessa função callback que são feitas as operações em cima dos pacotes.

Com a utilização da LipPcap, o sistema conseguiu ser capaz de fazer a captura de pacotes de uma determinada interface, com a possibilidade de adição de filtros à captura.

É importante ressaltar o funcionamento em baixo-nível da LibPcap, que torna necessário fluência no uso de ponteiros da linguagem C. Visto que todo o trabalho com as estruturas que representam os pacotes (cabeçalhos e dados) e que devem ser explicitamente declaradas no código, são utilizados ponteiros.

Por fim, também é interessante que o desenvolvedor faça algumas validações nos pacotes e nos seus cabeçalhos, que podem ajudar bastante na depuração de problemas.

5.3 Algoritmo para detecção dos ataques

A princípio, estudos teóricos foram feitos nos ataques a serem detectados (PortScan e SYN Flood), e surgiram algoritmos para a detecção dos mesmos.

Para a aplicação dos dois algortimos no sistema, foi necessário desenvolver uma maneira de manter uma lista dos últimos N pacotes que foram capturados, onde N pode ser modificado diretamente no código do programa.

A partir dessa lista dos últimos pacotes que foram recebidos, os algortimos podem detectar seus respectivos padrões e alertar caso haja uma suspeita de ataque.

5.4 Registro de visualização dos eventos

O IDS faz o registro dos eventos no momento que eles ocorrem, escrevendo a saída com as informações pertinente em um arquivo. Tal arquivo já é gerado como padrão de entrada para a ferremente de visualização Afterglow [21].

Assim, o adminitrador consegue ter um relatório em texto e em modo visual das detecções.

5.5 Alertas

Utilizando scripts em BASH e o cliente SMTP Email [22], o sistema é capaz de enviar um alerta ao administrador no momento que detecta um ataque. No alerta não são contidos nenhum tipo de informação sobre o ataque em si, apenas consta um aviso para o adminitrador verificar os relatórios visuais ou em texto.

6 Resultados

7 Considerações finais e trabalhos futuros

Referências

1. J. F. KUROSE, K. W. ROSS, **Redes de Computadores e Internet.** São Paulo: Ed. Addison Wesley, 3a. ed., 2006
2. SCHILDT, Herbert. **C Completo e total.** São Paulo: Ed. Makron Books, 3a. ed., 1996.
3. FUSCO, J. **The Linux Programmer's Toolbox**. Stoughton: Prentice-Hall, 2007.
4. TANENBAUM, Andrew S. **Redes de computadores.** Rio de Janeiro: Ed. Campus, 2003.
5. COMER, Douglas. **Interligação em rede com TCP/IP**. Rio de Janeiro: Ed. Campus, 1999.
6. COMER, Douglas. **Redes de computadores e Internet**. Porto Alegre: Ed. Bookman, 2001.
7. NOGUEIRA, J. P. Tiago. **Invasão de Redes:** Ataques e Defesas. Rio de Janeiro: Ed. Ciência Moderna, 2005.
8. ASSUNÇÃO, F. A. Marcos. **Honeypots e Honeynets.** Florianópolis: Ed. Visual Books, 2009.
9. RAYMOND, S, Eric. **The New Hacker's Dictionary.** United States of America: The MIT Press, 3 rd ed, 1996.
10. K. SCARFONE, P. MELL. **Guide to Intrusion Detection and Prevention Systems (IDPS).** 2010. Disponível em: <http://csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf>. Acesso em: 25 de Outubro de 2012.
11. K. H. SILVEIRA. **Desafios para os Sistemas de Detecção de Intrusos (IDS).** 2000. Disponível em: <http://www.rnp.br/newsgen/0011/ids.html>. Acesso em: 25 de Outubro de 2012.

12. V. SANTOS. **Sistemas de Detecção de Intrusões (IDS – Intrusion Detection Systems) usando unicamente softwares Open Source**. 2010. Disponível em: <http://www.seginfo.com.br/sistemas-de-deteccao-de-intrusoes-ids-intrusion-detection-systems-usando-unicamente-softwares-open-source/>. Acesso em: 25 de Outubro de 2012.
13. USAGE statistics and market share of Linux for websites. 2012. Disponível em: < http://w3techs.com/technologies/details/os-linux/all/all >. Acesso em: 23 de Julho de 2012.
14. DAVIX. 2012. Disponível em: <http://secviz.org/content/the-davix-live-cd>. Acesso em: 23 de Julho de 2012
15. NETFILTER. 2012. Disponível em: <http://www.netfilter.org/>. Acesso em: 23 de Julho de 2012.
16. IPFW. 2012. Disponível em: <http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/firewalls-ipfw.html>. Acesso em: 23 de Julho de 2012.
17. SNORT. 2012. Disponível em: <http://www.snort.org>. Acesso em: 23 de Julho de 2012.
18. SECVIZ. 2012. Disponível em: <http://www.secviz.org>. Acesso em: 23 de Julho de 2012.
19. SYN FLOOD. 2012. Disponível em: <http://pt.wikipedia.org/wiki/Syn_flood>. Acesso em: 25 de Outubro de 2012.
20. TCPDUMP/LIBPCAP. 2012. Disponível em: <http://www.tcpdump.org>. Acesso em: 25 de Outubro de 2012.
21. AFTERGLOW. 2012. Disponível em: <http://sourceforge.net/projects/afterglow/>. Acesso em: 25 de Outrubro de 2012.
22. EMAIL. 2012. Disponível em: <http://www.cleancode.org/projects/email>. Acesso em 25 de Outubro de 2012.