Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

ALEFF MARTINS DOS SANTOS

ANÁLISE DE FERRAMENTAS IDS (INSTRUCTION DETECTION SYSTEM) PARA PREVENÇÃO DE ANOMALIAS

Assis/SP
2017

Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

ALEFF MARTINS DOS SANTOS

ANÁLISE DE FERRAMENTAS IDS (INSTRUCTION DETECTION SYSTEM) PARA PREVENÇÃO DE ANOMALIAS

Projeto de pesquisa apresentado ao curso de Ciências da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientando(a): Aleff Martins dos Santos
Orientador(a): Prof. Me. Fábio Eder Cardoso

Assis/SP
2017

FICHA CATALOGRÁFICA

S247a SANTOS, Aleff Martins dos
Análise de ferramentas IDS (instruction detection system) para prevenção anomalias / Aleff Martins dos Santos . – Assis,2017.

44p.

Trabalho de conclusão do curso (Ciência da Computação). – Fundação Educacional do Município de Assis-FEMA

Orientador: Ms. Fábio Éder Cardoso

1.IDS 2.Segurança

CDD 005.8

ANÁLISE DE FERRAMENTAS IDS (INSTRUCTION DETECTION SYSTEM) PARA PREVENÇÃO DE ANOMALIAS

ALEFF MARTINS DOS SANTOS

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: Me. Fábio Eder Cardoso

Examinador: Prof. Douglas Sanches da Cunha

Assis/SP
2017

DEDICATÓRIA

Dedico este trabalho a minha família e ao orientador Fábio Eder pela sua compreensão, auxilio e sabedoria e a todos as pessoas que me ajudaram em minha jornada.

AGRADECIMENTO

Agradeço a Deus que me deu sabedoria para adquirir conhecimentos necessários para enfrentar todas as dificuldades e obstáculos encontrados no caminho.

A minha família que apesar das dificuldades me fortaleceu e que para mim foi muito importante.

Ao meu orientador, Fábio Eder Cardoso, pela oportunidade e apoio na elaboração deste trabalho.

E a todos que direta ou indiretamente fizeram parte da minha formação , o meu muito obrigado.

Tenho a impressão de ter sido uma criança brincando à beira-mar, divertindo-me em descobrir uma pedrinha mais lisa ou uma concha mais bonita que a outra, enquanto o imenso oceano de verdade continua misterioso diante de meus olhos.

Isaac Newton

(1643-1727)

RESUMO

O uso constante da transmissão de dados e o aumento da conectividade dos dispositivos por meio da Internet expõe os mesmos a exploração de vulnerabilidades em ambos os setores públicos e privados.

Cabe ao administrador de redes garantir a segurança e a integridade desses dispositivos por meio da análise do trafego de redes para que as informações não sejam expostas. Mas como o envio de informações acabam sendo excessivas o administrador muitas vezes acaba não acompanhando o tráfego anômalo.

A utilização de um sistema de detecção de intrusão (IDS), pode ser a solução que o administrador precisava para garantir a segurança e a integridade dos dados que trafegam em sua rede.

Palavras-chave: IDS, rede, segurança, dados, vulnerabilidades.

ABSTRACT

The constant use of data transmission and the increased use of Internet in devices by means of the Internet exposes them to exploitation of vulnerabilities in both the public and private sectors.

It is up to the network administrator to ensure the security and integrity of these devices by making use of the traffic analytics in the networks so that the information will not be exposed. But as the sending of information end up being excessive, the administrator often ends up not keeping up with the anomalous traffic.

The use of an intrusion detection system (IDS), may be the solution: the administrator may want to ensure the security and integrity of data that travels on his network.

Keywords: IDS, network, security, data, vulnerabilitys.

Lista de ilustrações

LISTA DE ABREVIATURAS E SIGLAS

IDS – INTRUSION DETECTION SYSTEM
TCP - TRANSMISSION CONTROL PROTOCOL
IP – INTERNET PROTOCOL
DARPA - DEFENSE ADVANCED RESEARCH PROJECTS AGENCY
HTTP – HIPERTEXT TRANSFER PROTOCOL
FTP – FILE TRANSFER PROTOCOL
SMTP – SIMPLE MAIL TRANSFER PROTOCOL
UDP – USER DATAGRAM PROTOCOL
NIDS - NETWORK INTRUSION DETECTION SYSTEM
HIDS - HOST INTRUSION DETECTION SYSTEM
OISF - OPEN INFORMATION SECURITY FUNDATION
WAF - WEB APPLICATION FIREWALL
CRS – CORE RULE SET
GB – GIGABYTE
GB - MEGABYTE
RAM – RANDOM ACCESS MEMORY
HD – HARD DISK
LAMP – LINUX, APACHE, MYSQL/MARIADB, PHP/PYTHON
SQL – STRUCTURED QUERY LANGUAGE
PHP – HYPERTEXT PREPROCESSOR
GNU – GNU'S NOT UNIX
CGI - C OMMON GATEWAY INTERFACE
WAF - WEB APPLICATION FIREWALL
SQLi - SQL INJECTION
XSS - CROSS SITE SCRIPTING
LFI - LOCAL FILE INCLUSION
RFI - REMOTE FILE INCLUSION
RCE - REMOTE CODE EXECUTION
OWASP - OPEN WEB APLICATION SECURITY PROJECT
CRS - CORE RULE SET

SUMÁRIO

1. INTRODUÇÃO

A internet surgiu na década de 1970 e 1980, como meio de comunicação entre os soldados que participaram da guerra fria em caso de um ataque do inimigo. Nessa década a internet também era utilizada para fins acadêmicos , as universidades dos estados unidos utilizavam links de rede mundial para trocar mensagens entre si.

Em 1990 o engenheiro inglês Tim Bernes-Lee desenvolveu uma interface gráfica para a internet a famosa *World Wide Web* (Rede Mundial de Computadores), ocasionando o crescimento em ritmo acelerado da web. E com o surgimento das redes sociais em 2006 e os sites de compra coletiva 2010, a internet se torna o principal meio de comunicação entre a população mundial, empresas e instituições acadêmicas.

Com o uso constante da comunicação de dados e o aumento da conectividade dos dispositivos por meio da Internet expõe esses dispositivos a exploração de vulnerabilidades tanto no meio corporativo quanto no meio institucional.

Cabe ao administrador de redes garantir a segurança e a integridade desses dispositivos por meio da análise do tráfego de redes para que as informações não sejam expostas. Mas como o envio de informações acabam sendo excessivas o administrador muitas vezes acaba não acompanhando o tráfego anômalo necessitando de ferramentas que o auxiliam na detecção das dessas ameaças.

Segundo KIZZA (2005) e CHAVE(2002) Sistemas de Detecção de Intrusão (IDS – *intrusion Detection System*) são ferramentas que visam melhorar a segurança em um sistema computacional. Detecção de Intrusão são técnicas utilizadas para detectar ataques ou perturbações a um sistema computacional ou rede de computadores .

KIM e REDDY(2005) reforçam a ideia de que o uso de ferramentas para a analise e monitoramento do trafego desempenham um papel importante no contexto de gerenciamento de segurança da rede.

A aplicação de um IDS que trabalhe em tempo real em uma rede local pode ser a garantia de uma rede segura. Uma vez que a vulnerabilidade é descoberta a ferramenta envia um alerta ao administrador de redes de uma possível falha encontrada.

1.1. MOTIVAÇÃO

Com o surgimento das redes sociais, sites de compra coletiva e smartphones a internet se tornou o principal vinculo de comunicação e troca de dados entre a população mundial, com isso ocasionando aumento eminente do tráfego malicioso.

A ideia de se estudar um software que faça o monitoramento da rede local em tempo real é bem interessante, pois engloba várias técnicas e conhecimentos presentes na área de redes de computadores, além de contribuir com os profissionais da área com o conhecimento adquirido neste trabalho.

1.2. OBJETIVO

O presente trabalho tem como objetivo principal, o estudo e analise do Sistemas de Detecção de Intrusão modsecurity para a defesa contra a invasão de servidores em uma instituição de ensino por meio de ferramentas de *penetration testing* (teste de intrusão) como por exemplo o sqlmap.

1.3. JUSTIFICATIVA

Segundo Tanembaum (2003, p.767), segurança é um tema que abrange diversas peculiaridades, que envolve diversos tipos de problemas, a preocupação em não deixar pessoas mal-intencionadas lerem ou modificarem mensagens enviadas a outros destinatários; acesso remoto por pessoas não autorizadas também estão incluídas neste contexto.

Pensando na segurança da instituição na qual eu estudo, os conceitos e software apresentados neste trabalho sobre segurança da informação visam apresentar métodos de como garantir a segurança dos dados que trafegam na rede utilizando ferramentas *open source* para fazer o monitoramento e prevenção da rede.

1.4. ORGANIZAÇÃO DO TRABALHO

O presente trabalho foi organizado em 6 capítulos, sendo o primeiro está introdução .

No capitulo 2 é explicado o que é segurança da informação e no que ela é utilizada, neste mesmo capitulo é explicado o que são conjuntos de protocolos TCP/IP, o que são IDS's e são mostrado quatro exemplos de ferramentas bastante utilizadas no mercado de segurança.

No capitulo 3 será explicado como o sistema detecção de invasão será instalado no computador para fazer uma tentativa de invasão ao servidor de aplicação para se testar a eficacia de um IDS.

No capítulo 4 é apresentado os resultados obtidos durante o teste.

O capitulo 5 apresenta as conclusões e o capitulo 6 os trabalhos futuros referente a esta pesquisa.

E por fim o capítulo 7 contém as referências utilizadas neste trabalho.

2. SEGURANÇA DA INFORMAÇÃO

Segundo o dicionário Aurélio, informação é o conjunto de dados acerca de alguém ou de algo. Estendendo esse conceito, podemos dizer que a informação é a interpretação desses dados. De nada vale um conjunto de dados sem que se faça a interpretação dos mesmos para se extrair um conhecimento útil.

> “Pouco nos adiantará se conhecermos métodos de ataques e proteção de um sistema ou rede se não conhecermos os princípios básicos de segurança da informação (GIAVAROTO; SANTOS, 2013, p.31).”

A segurança da informação se baseia em três princípios básicos que são:

Princípio da confidencialidade: este principio determina que apenas pessoas autorizadas possam ter acesso a determinadas informações. O acesso a um computador de forma “bruta” (por quebra de senha) para ser obter determinada informação seria uma violação deste princípio (GIAVAROTO; SANTOS, 2013, p.7).

Princípio da Integridade: determina que uma informação que seja integra e que não foi modificada pode ser considerado uma informação confiável. A informação perde sua integridade a partir do momento que for alterada intencionalmente ou não. Exemplo: o aluno que tenta alterar sua média na escola de 4 para 10 em um sistema de notas estará adulterando a informação de forma intencional levando a informação perder a sua confiabilidade (GIAVAROTO; SANTOS, 2013, p.31).

Princípio da Disponibilidade: define que a informação esteja sempre disponível a pessoa autorizada. Podemos considerar como violação deste principio um ataque de negação de serviço contra o servidor forçando - o a parar de funcionar levando as informações que se encontram nele a ficarem indisponíveis (GIAVAROTO; SANTOS, 2013, p.31).

2.1. PROTOCOLOS TCP/IP

De acordo com ASSUNÇÃO, o conjunto de protocolos TCP/IP foi criado pela *DARPA (Defense Advanced Research Projects Agency)* para fornecer a

comunicação através da agencia. Os Estados Unidos da América queriam uma rede que pudesse sobreviver a qualquer guerra ou conflito, então criaram o TCP/IP, para garantir que os pacotes sempre chega-se ao seu objeto.

Para entendermos melhor como o TCP/IP funciona, imagine a seguinte situação,se você pudesse viajar pelo mundo e fosse para Inglaterra e quisesse se comunicar com a população local teria que saber falar em inglês, ou se fosse para Rússia ou o Japão, mesma situação você teria que saber falar russo e japonês causo queira se comunicar, certo? Não, bastaria você saber falar inglês fluentemente que você poderia ir para qualquer canto do mundo, e se comunicar com qualquer pessoa sem o menor problema já que o inglês é o idioma mais falado no mundo. A agora imaginem que a um monte de computadores espalhados pelo mundo, com fabricantes, arquitetura e sistemas operacionais diferentes, como fazer eles se conversarem entre si? Eis que surge os protocolos TCP/IP, que funciona como uma “linguagem universal” para os computadores se comunicarem entre si (Olonca, 2007).

Os protocolos TCP/IP são divididos em quatro camadas diferentes, estes são:

Camada de Aplicação: esta camada contém os protocolos de alto nível (HTTP, FTP, SMTP, entre outros). Todas as operações com esses procolos bem como suas propriedades, sessões e controle de dialogo são realizados nessa camada. Após o termino do processo, os dados são empacotados e encaminhados para o próxima camada (ASSUNÇÃO, 2012).

Camada de Transporte: responsável pelo controle de fluxo, confiabilidade e possíveis correções de erros na entrega de dados. Nessa camada encontramos os protocolos TCP e UDP (ASSUNÇÃO, 2012).

Camada de Internet: está camada possui a função de assegurar que os dados cheguem ao seu destino, independente do caminho (rotas) e das redes utilizadas para isso. O protocolo responsável por gerenciar está camada é o protocolo IP (Internet Protocol, que traduzido para o português fica Protocolo de Internet) (ASSUNÇÃO, 2012).

Camada de Rede ou Host-Rede: a tarefa desta camada é receber e enviar pacotes pela rede. Os protocolos utilizados nessa camada podem variar

dependendo do tipo de rede que está sendo utilizado. Atualmente, o mais comum é o Ethernet (ASSUNÇÃO, 2012).

Logo abaixo é apresentado uma imagem ilustrando os protocolos e redes existentes no TCP/IP na visão de TANENBAUM.

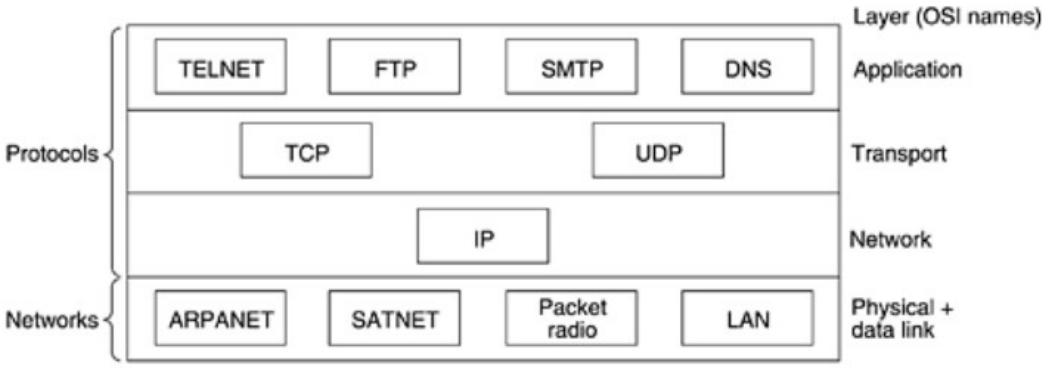

Figura 1: Protocolos e Redes no TCP/IP (In: TANEMBAUM et al., 2003, p. 49)

2.2. SISTEMAS DE DETECÇÃO DE INTRUSÃO

As IDS são sistemas de detecção de invasão em tempo real. Basicamente o que elas fazem é alertar o administrador de redes de uma possível tentativa de invasão em sua rede, em alguns casos essas ferramentas podem de forma reativa tomar certas ações contra o ataque. Os IDS's são divididos em dois tipos de sistemas que são as NIDS e as HIDS:

De acordo com SANTOS (2010) NIDS (*Network Intrusion Detection System* – Sistemas de Detecção de Intrusão em Rede) são ferramentas que analisam o conteúdo dos pacotes que trafegam na rede. A imagem abaixo ilustra o funcionamento de um sistema de detecção de intrusão em rede.

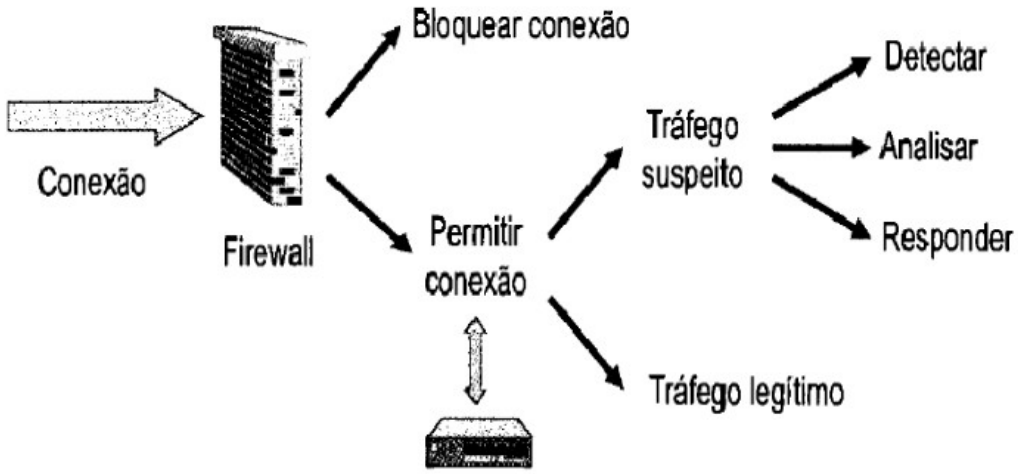

Figura 2: Função dos NIDS'S.
Fonte: Análise dos Sistemas de Detecção de Intrusão em Redes: Snort e Suricata comparando com dados da Darpa - Cléber Taschetto Murini.

Como é mostrado na figura acima após as conexões serem liberadas pelo *firewall*, as informações são coletadas pelo sistema de detecção para serem armazenadas e analisadas, para informar se o tráfego é legitimo ou suspeito.

De acordo com MURINI, Para ter um bom desempenho na detecção de intrusão o sistema necessita de algumas ferramentas básicas, explicadas logo abaixo:

O coletor o qual é responsável por coletar os pacotes que trafegam pela rede, geralmente ele é instalado depois do *firewall* (MURINI, 2014).

O analisador é responsável por verificar em busca de alguma anomalia os pacotes capturados pelo coletor (MURINI, 2014).

O banco de dados é aonde são armazenados as informações dos NIDS e os eventos suspeitos, para busca e uma analise mais detalhada quando necessário.

Notificador manda um alerta ao administrador de redes causo entre alguma ameça seja encontrada (MURINI, 2014).

Atuador: possui a capacidade de tomar ações automatizadas quando alguma ameça é encontrada(MURINI, 2014).

Segundo SANTOS (2010) HIDS *(Host Intrusion Detection System* – Sistema de Detecção em Host) são sistemas que possuem a capacidade de examinar ações baseados em host (computador) especifico, bem como os aplicativos e arquivos

estão sendo usados quais arquivos entre outras funções. A seguir segue uma imagem ilustrando o funcionamento de um HIDS.

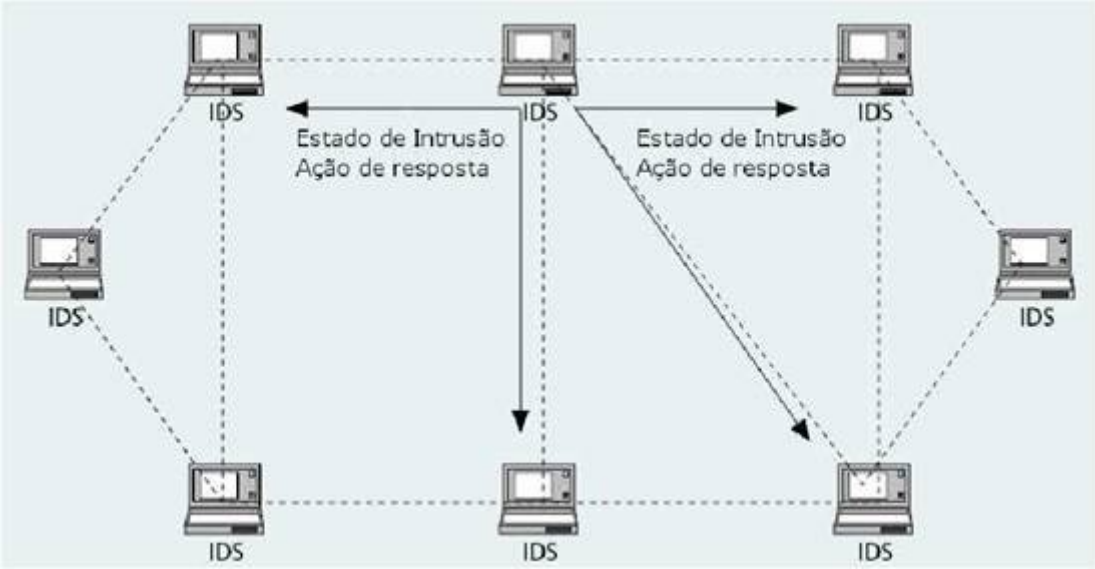

Figura 3: Funcionamento de um HIDS.
Fonte: http://www.devmedia.com.br/sistema-de-deteccao-de-intrusao-artigo-revista-infra-magazine-1/20819

2.2.1. FORMAS DE DETECÇÃO

A seguir serão explicadas todas as formas de detecção presentes nos sistemas de detecção de intrusão.

- **Detecção por Assinatura:** analisa as atividades do sistema procurando por eventos que correspondam a padrões (assinaturas) pré-definidos de ataques e atividades maliciosas. Uma desvantagem dessa forma de detecção é que este tipo de técnica só detecta ataques conhecidos, necessitando assim de constante atualização (SANTOS, 2010).

- **Detecção por Anomalia:** parte do principio de são ações diferentes das atividades normais de sistemas, ou seja, qualquer atividade que o sistema ou usuário realize fora padrão o IDS alertará o administrador da

possível ameaça. Uma desvantagens é que o sistema pode gerar muitos alertas falsos devido ao comportamento imprevisível do usuário (SANTOS, 2010).

- **Detecção baseada em especificação:** este modelo atua logo abaixo da camada de aplicação de pilha de procolos ou no nível de controle do sistema operacional. Esse modelo se restringe à execução correta de um programa ou protocolo monitorando o seu funcionamento (SANTOS, 2010).

A seguir serão apresentadas alguns sistemas de segurança de intrusão open sources .

2.2.2. SURICATA

O suricata é um motor de alto desempenho de Rede, IDS, IPS e Monitoramento de Segurança de Rede. Ele é de código aberto e de propriedade de uma fundação sem fins lucrativos administrada pela comunidade, a OISF (Suricata, 2017).

De acordo com Cleber Murini (2014), o suricata é uma ferramenta multi-threaded capaz de equilibra a carga de processamento em cada processador em um sensor, permitindo que o hardware *commodity* alcance velocidades de até 10 *Gigabit* sobre o tráfego da vida real, sem sacrificar a cobertura do conjunto de regras.

O suricata é baseado em detecção por assinatura ou seja ele compara os dados coletados com uma base de dados de ataques ou regras pré-definidas. A IDS possui um motor capaz de fazer a detecção de intrusão em tempo em tempo real e monitoramento de segurança de rede, sendo capaz também de trabalhar no processamento *off-line* de pcap (captura de fichas) (Suricata, 2017).

2.2.3. SNORT

O Snort é uma das ferramentas mais utilizadas atualmente em servidores, ele é mantido por Brian Caswell e Marty Roesch. O snort possui versões para Linux, Windows, FreeBSD ambas as versões podem ser abaixadas direto do site oficial do snort.

De acordo com MURINI (2014), o snort se baseia na aplicação de regras de filtragem em cada pacote, a fim de procurar assinaturas de ataques conhecidos. E que a limitação dessa ferramenta se refere ao ato de apenas procurar assinaturas de ataques em cada pacote, ou em um fluxo, sem conseguir detectar ataques que necessitem de pacotes de protocolos diversos para serem caracterizados.

A seguir é mostrado uma ilustração mostrando o fluxo do snort:

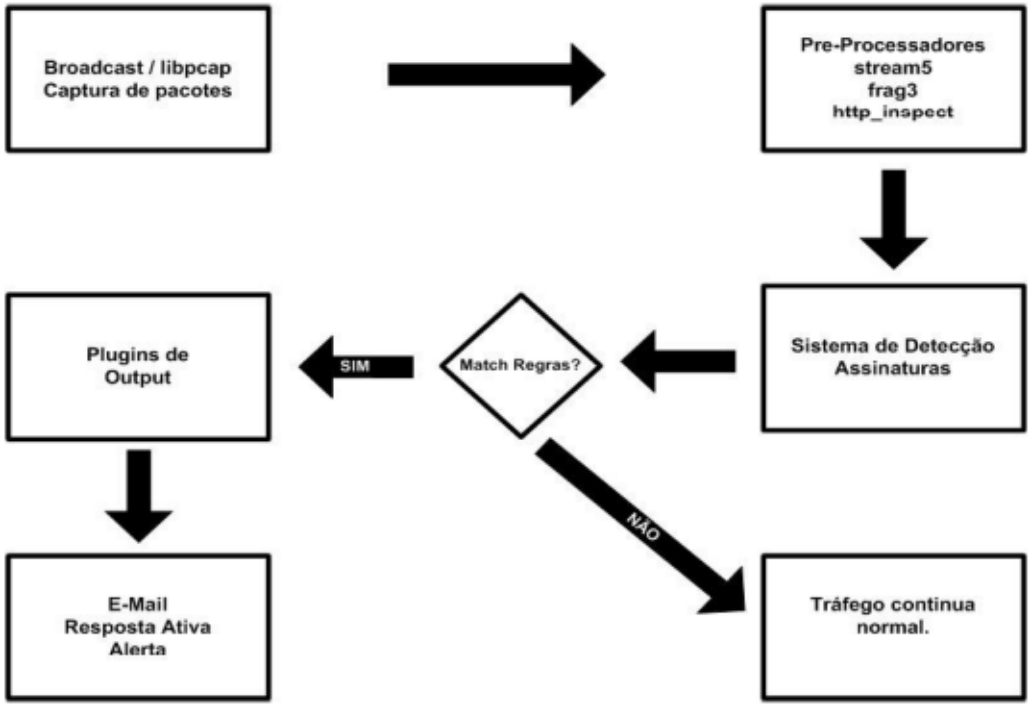

Figura 4: Fluxo do Snort.
Fonte: https://seginfo.com.br/2010/06/21/sistemas-de-deteccao-de-intrusoes-ids-intrusion-detection-systems-usando-unicamente-softwares-open-source/

O snort é capaz de fazer a detecção de ataques do tipo:

- *buffer overflow:* é o resultado do armazenamento em *buffer*[1] de quantidade superior de dados que do que sua capacidade (ARANHA, 2003).
- *stealth port scans*: é uma técnica usada para detectar que portas estão abertas em um computador. Baseado nas informações sobre as portas abertas, o acesso não autorizado pode ser obtido (SYMANTEC, 2017).
- ataques CGI (*Common Gateway Interface*): por meio deste ataque o invasor consegue modificar o conteúdo e na consequente degradação da imagem da empresa (NAKAMURA ; GEUS, 2007, p. 124).

2.2.4. MODSECURITY

A *engine* é suportado pela a equipe da SpiderLabs, e mantido pela TrusWave um empresa bem conhecida na área de segurança.O ModSecurity é um modulo de firewell de aplicação web (WAF) de código aberto e opensource, ele é capaz de monitorar o tráfego HTTP em tempo real para detecção na Web (HORA, 2017).

De acordo com TrusWave, os WAFs são implantados para estabelecer uma camada de segurança externa aumentada para detectar e / ou prevenir ataques antes que eles atinjam aplicativos da Web. O ModSecurity fornece proteção contra uma série de ataques contra aplicativos da Web e permite o monitoramento de tráfego HTTP e análise em tempo real com poucas ou nenhuma mudança na infra-estrutura existente.

A seguir é mostrado uma imagem do ModSecurity implementado na Rede.

[1] Local onde se armazena dados temporários que são movidos de um local para o outro.

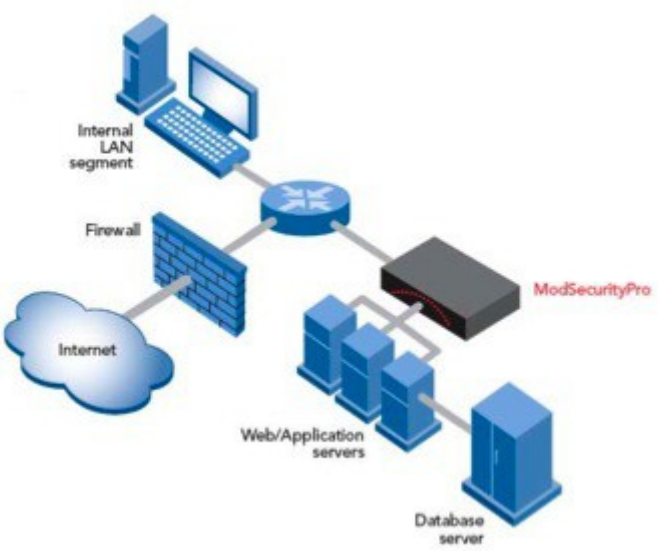

Figura 5: Função do ModSecurity.
Fonte: https://www.helpnetsecurity.com/2007/08/02/new-web-application-firewall-appliance-based-on-modsecurity/

A imagem acima ilustra como o modsecurity de ser implementado deve ser implementado na rede, logo depois do *firewall* e antes do servidores de aplicação e banco de dados para que o mesmo possa capturar e analisar as requisições que são feitas tanta da rede local quanto da rede externa,

O SpiderLabs da Trustwave criou o Projeto de Conjunto de Regras Principais (CRS) do OWASP ModSecurity. Ao contrário dos sistemas de detecção e prevenção de intrusões, que dependem de assinaturas específicas para vulnerabilidades conhecidas, o CRS fornece proteção genérica contra vulnerabilidades desconhecidas freqüentemente encontradas em aplicativos web (HORA, 2017).

Estas são alguns ataques que o conjunto de regras básicas fornece proteção:

- SQL Injection (SQLi);
- Cross Site Scripting (XSS);
- Local File Inclusion (LFI);
- Remote File Inclusion (RFI);
- Remote Code Execution (RCE);

- PHP Code Injection;
- HTTP Protocol Violations;
- HTTPoxy;
- Shellshock;
- Session Fixation;
- Scanner Detection;
- Metadata/Error Leakages;
- Project Honey Pot Blacklist;
- GeoIP Country Blocking;

2.2.5. FAIL2BAN

O *Fail2Ban* é uma aplicação que analisa continuamente os ficheiros *log* e bloqueia os endereços Internet de onde originaram várias tentativas falhadas de acesso com *senha* inválida (DEBIAN, 2017).

Segundo FERREIRA (2016), o Fail2ban é capaz de proteger servidores contra ataques automatizados que fazem tentativas abusivas a determinado serviço. Apos detectar a anomalia, o Fail2ban irá adicionar uma nova regra no *firewall* iptables, bloqueando o endereço IP do atacante por um determinado período de tempo. Além disso, ele também poderá alertá o administrador de redes por e-mail sobre o que está ocorrendo.

3. METODOLOGIA

Para se testar os sistemas de detecção de intrusão,sera configurado um computador HP que foi cedido pela Fundação Educacional do Município de Assis.

3.1. IMPLEMENTAÇÃO

Foi cedido pela faculdade um computador HP o qual possui uma configuração razoavelmente boa para se fazer os teste com o IDS, estás são: memória de 1286 MB de RAM, HD de 40 GB no qual foi instalado o sistema operacional GNU/Linux Debian 8.8 e um processador Celeron .

O referido sistema operacional é muito bom para a configuração de servidores de acordo com a pesquisa feita pela W3Tech, uma em presa especializada em levantar dados de tecnologias utilizadas na rede, conforme ilustra a figura 6.

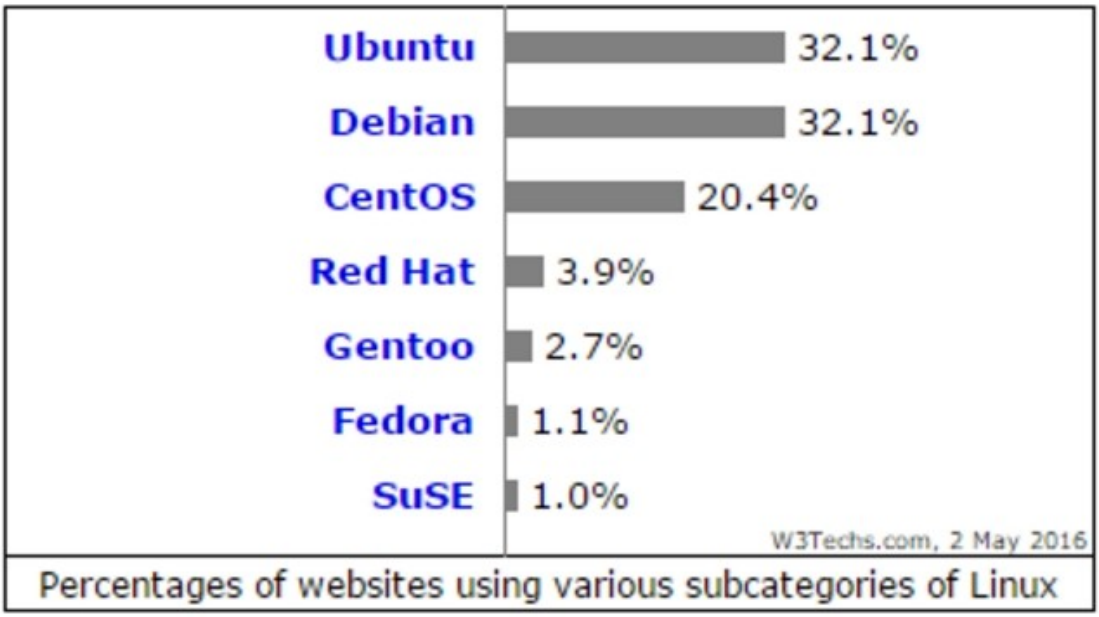

Figura 6: Porcentagem de sites utilizando várias distribuições linux.
Fonte: http://www.diolinux.com.br/2016/05/debian-e-ubuntu-sao-lideres-em-servidores.html

3.2. INSTALAÇÃO E CONFIGURAÇÃO DO LAMP

Será instalado junto com sistema operacional o conjunto de pacotes LAMP que é uma combinação de software livres e de código aberto. O acrônimo LAMP faz referência as primeiras letras de Linux (sistema operacional), Apache (servidor web), MariaDB ou MySQL (banco de dados) e PHP ou Python (linguagens de programação) (GUERRA, 2014).

É importante saber que antes de digitar os comandos de instalação do LAMP é necessário estar como usuário *root* do sistemas operacional que é indicado pelo simbolo de "#" *octohorpe* popularmente conhecido como sustenido, para não ter problemas com permissões (ALECRIM, 2005). Abaixo será listado uma serie de comandos que serão necessário para instalar o LAMP no Linux.

Para que a ferramenta IDS possa ser testada se faz necessário a instalação do banco de dados, servidor apache para poder executar a aplicação e a linguagem de programação PHP para desenvolver a aplicação. Ambas podem ser instalados através do comando descritos abaixo:

```
# apt-get install mysql-server mysql-client
# apt-get install apache2
# apt-get install php5 libapache2-mod-php5
```

Sempre que se instala o modulo PHP assim como os demais módulos é importante reiniciar o servidor apache para que o modulo possa ser reconhecido pelo servidor apache, para isto será utilizado o comando:

```
# service apache2 restart
```

Para se obter suporte ao PHP5 e ao MySQL será necessário a instalação dos seguintes pacotes, através do comando "*apt-get install*". Como novos módulos foram instalados no servidor, mais uma vez o mesmo necessita ser reinicializado para que as alterações realizadas nele possam ser aplicadas. Como mostrado nos comandos a seguir:

```
# apt-get install php5-mysql php5-curl php5-gd php5-intl php-pear php5-imagick
 php5-imap php5-mcrypt php5-memcache php5-pspell php5-pspell
 php5-recode php5-snmp php5-sqlite php5-tidy php5-xmlrpc php5-xsl
 # service apache2 restart
```

3.3. INSTALAÇÃO E CONFIGURAÇÃO DO MODSECURITY

O modsecurity foi selecionado para se garantir a segurança do servidor *web*. Pois ele é um *firewall* de aplicação que tem o objetivo de vários tipos de ataque, como por exemplo, SQL Injection, Cross-Site Scripting (XSS), Trojans, Backdoors Detection, dentre outros, que variam de acordo com as regras existentes (CEZAR, 2011).

A seguir será mostrado os comandos e as configurações necessárias para que o modsecurity possa funcionar corretamente no sistemas operacional Debian. Os seguintes pacotes são necessários para que o modsecurity possa ser utilizado:

```
# apt-get install apache2-threaded-dev libxml2-dev libcurl4-gnutls-dev liblua5.1-0
liblua5.1-0-dev build-essential php5-cli libghc-pcre-light-dev
```

O *download* pode ser feito por meio deste comando:

 # apt-get install zip libapache2-mod-security2 libxml2 libxml2-dev libxml2-utils libaprutil1 libaprutil1-dev

Parametrizando as configurações recomendadas do modsecurity:

mv /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/ modsecurity.conf

O servidor apache será reiniciado para que os módulos possam ser carregados:

service apache2 restart

Com os modulos do modsecurity carregados será feita a edição do arquivo de configuração onde será alterado a diretiva SecRuleEngine para On para que o modesecurity possa bloquear a possíveis ameaças enviadas ao servidor. Outra diretiva a ser modificar é SecResponseBodyAccess.

Quando se ativs essa diretiva, é realizado um buffer completo do conteúdo das requisições, esse *buffer* serve para que as regras do modsecurity possam fazer a leitura desses dados e decidir se libera ou não o conteúdo que entra no servidor. Aqui encontramos um problema, pois, o modsecurity costuma usar a memória RAM, para fazer esse *buffer*, o que pode degradar o desempenho do servidor, principalmente se estivermos compartilhando o modsecurity e o servidor *web* em um mesmo servidor. Abaixo será mostrado uma imagem de como arquivo deverá ficar depois de ser editado.

nano /etc/modsecurity/modsecurity.conf

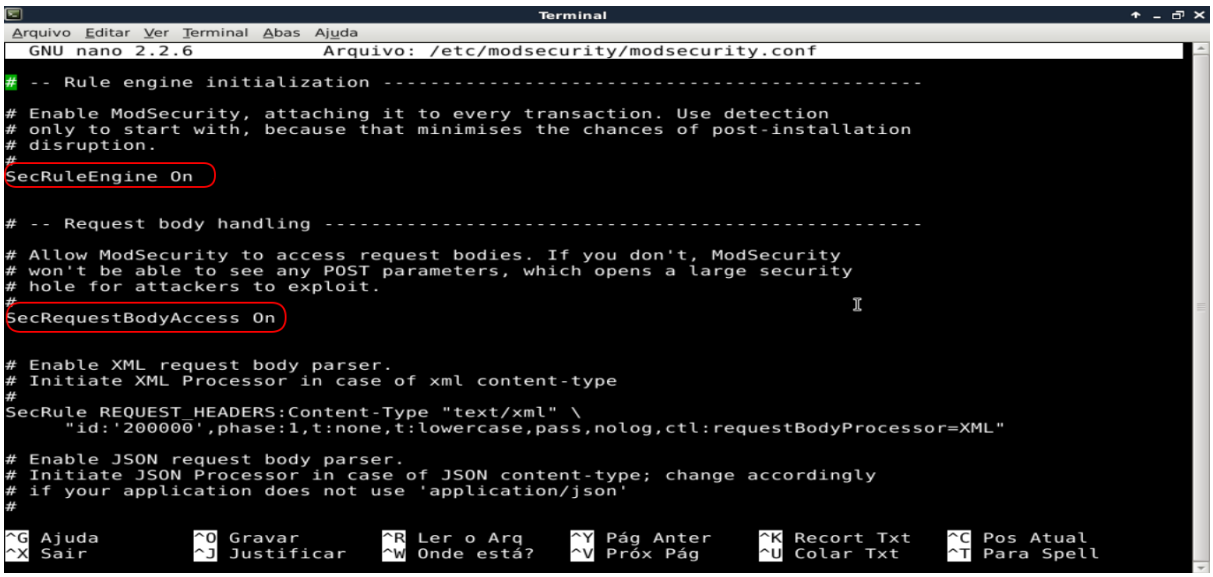

Figura 7: Arquivo de configuração do ModSecurity depois de editado.

Com modsecurity configurado será necessário que se faça o *download*[2] de sua base de dados que se encontra na versão 2.8. Será necessário descompactá-lo, utilizado o comando abaixo:

unzip owasp-modsecurity-crs-2.2.8.zip

Com o arquivo descompacto, é necessário que se faça *backup* da base do modsecurity caso precise refazê-la novamente, a copia dos arquivos será feita para o diretório etc (nesta pasta se encontra os arquivos de configuração e aplicativos do sistema) do Linux:

cp -R owasp-modsecurity-crs-2.2.8/* /etc/modsecurity/

Também se faz necessário a criação de *links* simbólicos para a pasta de regras ativas do modsecurity.

mv/etc/modsecurity/modsecurity_crs_10_setup.conf.example /etc/modsecurity/ modsecurity_crs_10_setup.conf

[2]https://github.com/SpiderLabs/owasp-modsecurity-crs/releases/tag/2.2.8

cd /etc/modsecurity

Para que o modsecurity possa consultar a sua base de dados é necessário dizer a ele em qual pasta elas se encontram, sendo assim basta adicionar o caminho da no arquivo security2.conf com seguinte comando em destaque.

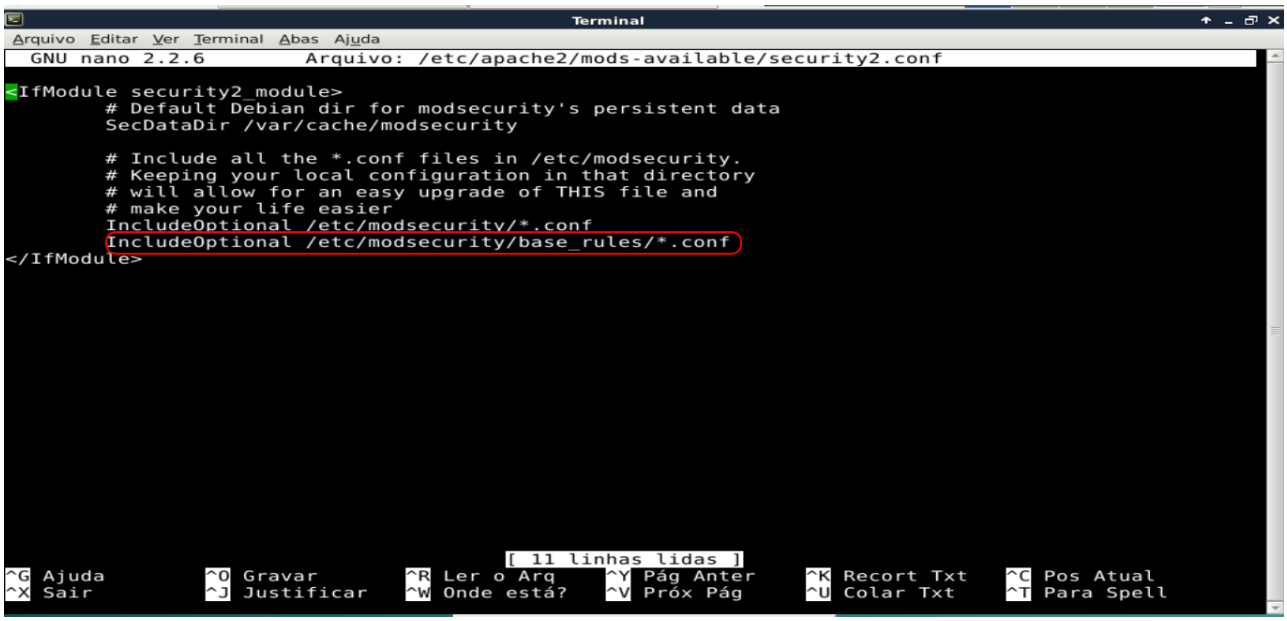

Figura 8: Arquivo security2.conf depois de editado.

Depois de feito as configurações será necessário habilitar os *headers* das novas configurações do apache, reiniciar o servidor para que as configurações possam ser aplicadas e subir as regras do modsecurity, com os seguintes comandos:

a2enmod headers

service apache2 restart

service apache2 reload

Será necessário que se faça a exclusão da base de dados do modsecurity para recriar o conteúdo da pasta somente com os módulos necessário para a detecção e bloqueio de sql injection. Para isto será utilizado os seguintes comandos:

```
# rm -r /etc/modsecurity/base_rules/*
# cp /root/Downloads/owasp-modsecurity-crs-2.2.8/base_rules/
  modsecurity_crs_41_sql_injection_attacks.conf /etc/modsecurity/base_rules/
# cp /root/Downloads/owasp-modsecurity-crs-2.2.8/base_rules/
  modsecurity_crs_41_xss_attacks.conf /etc/modsecurity/base_rules/
# cp /root/Downloads/owasp-modsecurity-crs-2.2.8/base_rules/
  modsecurity_crs_42_tight_security.conf /etc/modsecurity/base_rules/
# cp /root/Downloads/owasp-modsecurity-crs-2.2.8/base_rules/
  modsecurity_35_scanners.data /etc/modsecurity/base_rules/
```

Com os módulos copiados será necessário habilitar os headers, reiniciar o servidor e carregado a base de dados do modsecurity como mostrando no passo anterior.

4. TESTES REALIZADOS NO SERVIDOR

Com o modsecurity instalado e configurado no servidor chegou a hora de testar seu desempenho e eficacia no que diz respeito a detecção e bloqueio por meio de ataques de SQLi. Para que o teste pode-se ser feito foi colocado no servidor uma pequena aplicação vulnerável a injeção de SQL, como mostra as imagens a seguir:

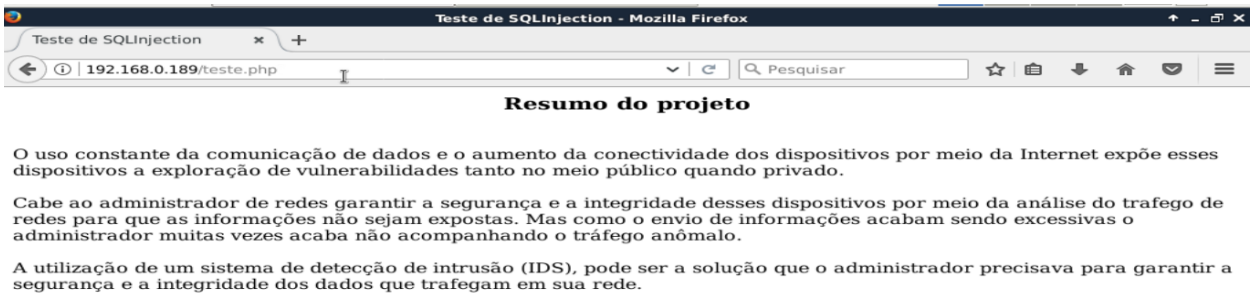

Figura 9: Aplicação feita em PHP.

```php
<?php
#  CONFIGURAÇÃO DE BANCO
$host = 'localhost';
$dbuser = 'root';
$dbpass = 'root@admin';
$dbname = 'blind';

echo "<title>Teste de SQLInjection</title>";
echo "<h3><center>Resumo do projeto</center></h3><br>";
echo "<left>O uso constante da comunicação de dados e o aumento da conectividade dos dispositivos por meio da Internet
 expõe esses dispositivos a exploração de vulnerabilidades tanto no meio público quando privado.</left><br><br>";
echo "<left>Cabe ao administrador de redes garantir a segurança e a integridade desses dispositivos por meio da análise do
 trafego de redes para que as informações não sejam expostas. Mas como o envio de informações acabam sendo excessivas o
administrador muitas vezes acaba não acompanhando o tráfego anômalo.</left><br><br>";
echo "<left>A utilização de um sistema de detecção de intrusão (IDS), pode ser a solução que o administrador precisava para
 garantir a segurança e a integridade dos dados que trafegam em sua rede.</left><br><br>";

$db = mysql_connect($host, $dbuser, $dbpass);
mysql_select_db($dbname,$db);
$sql = "SELECT * FROM users WHERE id=".$_GET['id'];
$query = mysql_query($sql);

if(@mysql_num_rows($query)==0){
      die;
}

$result=@mysql_fetch_row($query);

echo "<h3><center><u>Teste de SQL Injection </u><br><br>";
echo "<font color='#000000'>ID: </font>".$result[0]."<br>";
echo "<font color='#000000'>Usuario: </font>".$result[1]."<br>";
// echo "Password: ".$result[2]."<br>";
echo "</h2></center>";
die();
?>
```

Figura 10: Código da aplicação feita em PHP.

Os testes foram divididos em três etapas, no qual a primeira etapa consiste em realizar o ataque de SQLi utilizando o sistema operacional Xubuntu[3] GNU/Linux com o sqlmap que é uma ferramenta de teste de penetração de código aberto que automatiza o processo de detecção e exploração de falhas de injeção de SQL e a tomada de servidores de banco de dados. A segunda etapa foca-se em atacar o servidor com o modsecurity ativado e na terceira etapa será realizado um ataque ao servidor com o modsecurity e o *firewall* iptables trabalhando em conjunto para garantir a segurança a sua segurança e integridade.

4.1. RESULTADOS OBTIDOS

Neste tópico será mostrado os resultados adquiridos durante os ataques feito ao servidor durante cada etapa.

4.1.1. ETAPA 1

Como dito anteriormente neste trabalho a etapa 1 consiste em atacar um servidor desprotegido por meio da ferramente sqlmap, como o servidor não possui segurança, já era esperado que o ataque fosse bem sucedido, como lustra as imagens a seguir:

O comando abaixo foi utilizado para inserir a injeção sql no banco de dados através da aplicação teste.php e me trazer o nome do banco de dados, o resultado segue em destaque na imagem abaixo:

```
# sqlmap -u http://192.168.0.189/teste.php?id=1 - -current-db
```

[3] O Xubuntu é um procedente da distribuição Ubuntu GNU/Linux, ele vem com Xfce (ambiente de trabalho gráfico) além de programas que o façam rodar satisfatoriamente em computadores que possuem um hardware mais antigo.

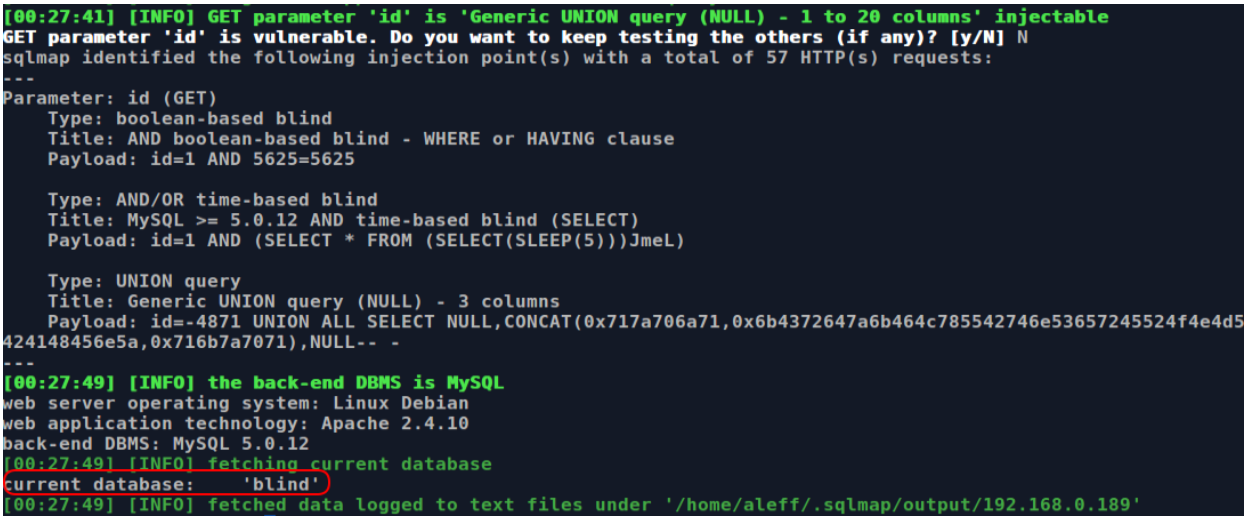

Figura 11: Banco de Dados listado através do ataque realizado com o sqlmap.

Com o banco de dados listado, deu-se continuidade ao ataque para se obter as tabelas do banco de dados através do comando:

sqlmap -u http://192.168.0.189/teste.php?id=1 -D blind –tables

Figura 12: Tabela listada através do ataque realizado com o sqlmap.

Com a tabela descoberta, foi utilizado o comando abaixo para se capturar as colunas da tabela:

sqlmap -u http://192.168.0.189/teste.php?id=1 -D blind -T users --columns

Figura 13: Colunas listadas através do ataque realizado com o sqlmap.

Com as colunas descobertas, o ataque segue para a etapa final onde se obtém os dados das colunas, através do comando:

sqlmap -u http://192.168.0.189/teste.php?id=1 -D blind -T users -C ‘id, name, password’ --dump

Figura 14: Dados das colunas listados através do ataque realizado com o sqlmap.

4.1.2. ETAPA 2

Como é mostrado no tópico 3.5.1, o servidor sem proteção alguma fica completamente vulnerável a ataques deste tipo. Neste tópico será feito os mesmos testes realizados na etapa 1, com o modsecurity instalado e

configurado no servidor. Um fato importante a se mencionar é que os teste feitos com o modsecurity foram realizados somente com sua base de dados gratuita. A seguir é mostrado algumas imagens tiradas durante os testes.

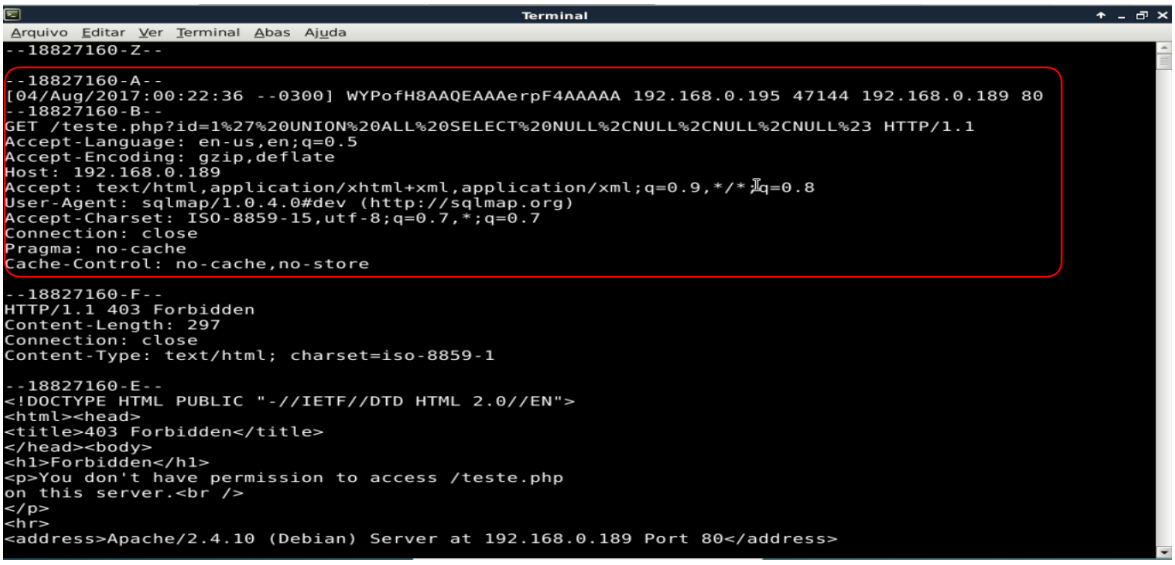

Figura 15: Arquivo de log do modsecurity após o ataque.

Como é mostrado na figura acima o modsecurity detectou que houve uma tentativa de injeção sql a partir do IP 192.168.0.195 (IP do atacante) ao servidor que é o endereço 192.168.0.189 pela porta 80 (porta do servidor apache) e fechou a conexão evitando a injeção de SQL no banco de dados. Abaixo é mostrado uma imagem da tela do atacante no momento que a injeção é bloqueada.

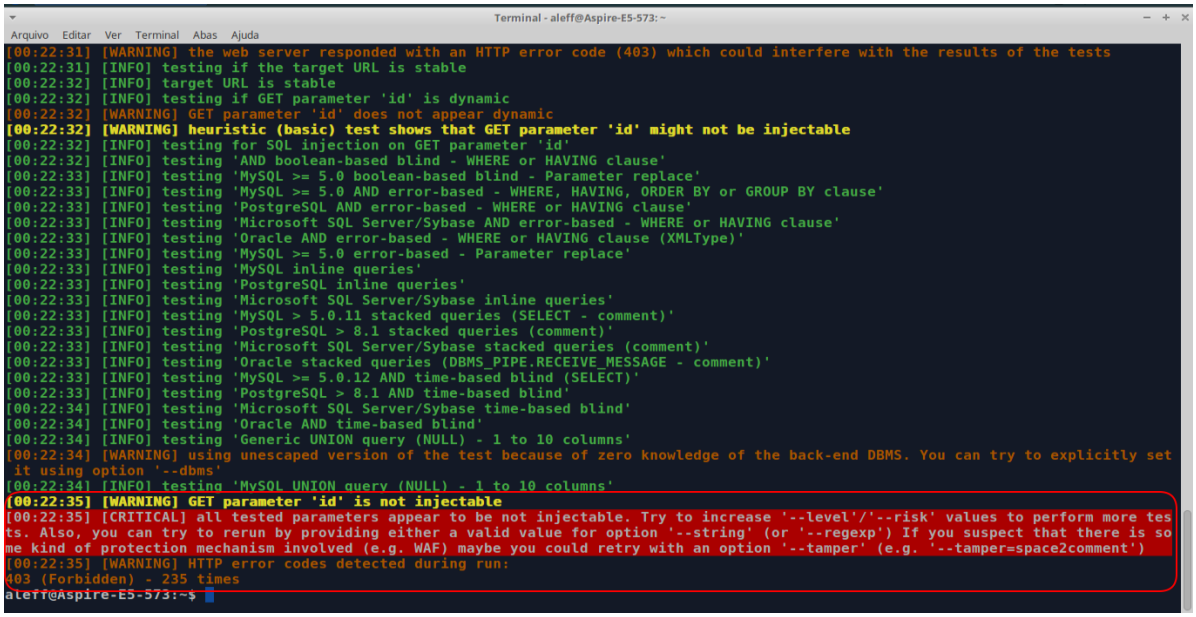

Figura 16: Segundo ataque de SQL injection ao servidor.

4.1.3. ETAPA 3

Nesta etapa será utilizado em conjunto com o modsecurity o firewall iptables que é um *firewall* em nível de pacotes, ou seja, ele funciona através da comparação de regras para saber se um pacote tem ou não permissão para passar.

Para este teste foram criadas algumas regras de *firewall* baseadas nas informas obtidas durante a analise do arquivo de *log* do modsecurity como ilustrado na figura 15. A seguir é mostra um *print screen* das regras de *firewall* criadas.

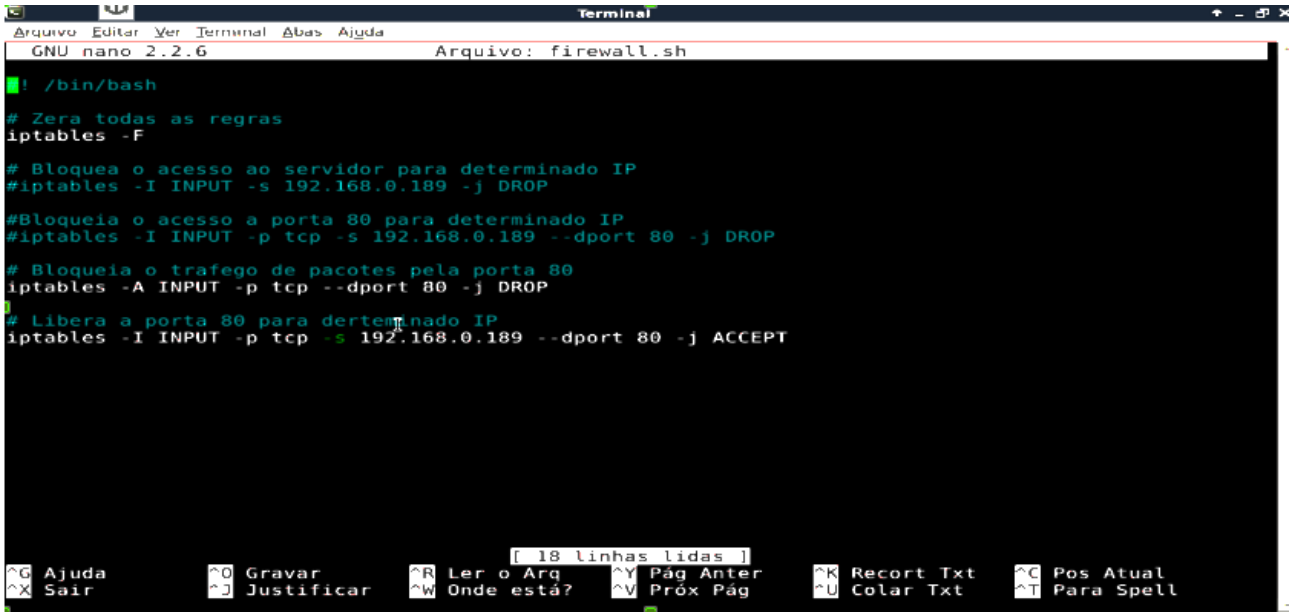

Figura 17: Regras de firewall.

Na imagem acima é mostras as regras utilizadas no *firewall* iptables para bloquear a entrada de pacotes na porta 80 do servidor e liberada passagem somente para o servidor se houve qualquer outro computador que queira se conectar ao servidor para fazer uso da aplicação não conseguiria a não ser que o administrador da rede desse permissão a ele pelo *firewall.*

Com as regras de *firewall* criadas, foi realizado o terceiro ataque ao servidor e como se era esperado o sqlmap não conseguiu fazer a injeção de sql pois a maquina que está fazendo o ataque não possui permissão no *firewall* como mostra a imagem a seguir.

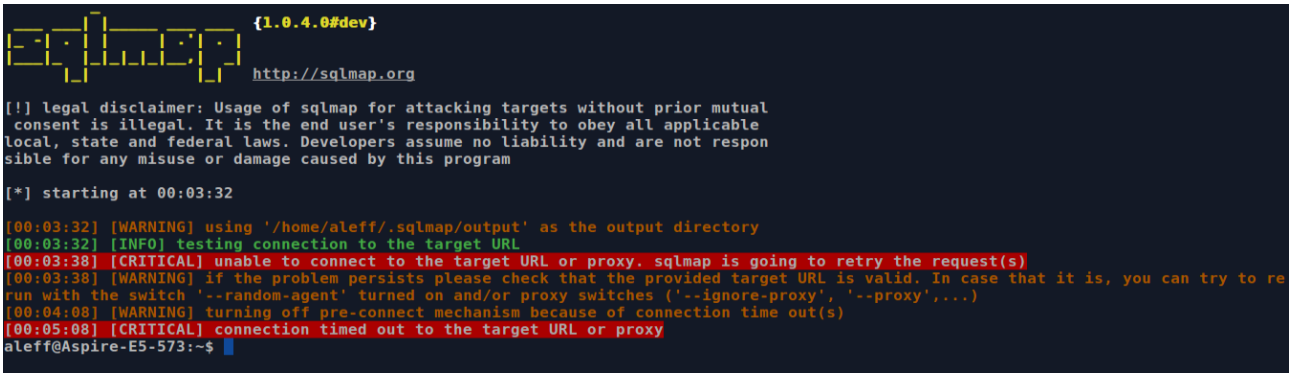

Figura 18: Ataque bloqueado pelo firewall iptables.

5. CONCLUSÃO

O modsecurity é uma excelente ferramenta de detecção de invasão, desde que seja configurada corretamente para evitar alertas de falsos/positivos e evitar a sobrecarregar o servidor, caso o IDS seja instalado junto do servidor de aplicação e do *firewall,* esse tipo de configuração eu só recomento para testes e estudos com ferramenta caso queira usá-la em um ambiente corporativa o mais recomenda seria seguir a configuração de rede mostrada na figura 5.

Contudo como já foi dito anteriormente apesar do *modsecurity* ser uma excelente ferramenta de detecção de intrusão assim como as outros IDS's, e possuir uma comunidade unida e ativa que está sempre cuidando e atualizado a sua base de dados *free a* constante evolução e aumento deste ataques tem ocasionados muitos problemas a todos no que se diz respeito a segurança.

A verdade é que não conseguimos evoluir as ferramentas de defesa na mesma velocidade que um *cracker*[4] evolui sua ferramenta de ataque ou planeja o seu próximo ataque. Sempre devemos trabalhar para evoluir os IDS's assim como criar novas formas de defesa contra esses ataques. Pois os sistemas de detecção de intrusão são de grande ajuda para o profissional na área de redes como foi comprovado durante o trabalho.

[4] Os crackers são pessoas que utilizam seu grande conhecimento na área de informática para quebrar códigos de segurança, senhas de acesso a redes e códigos de programas com fins criminosos.

6. TRABALHOS FUTUROS

Com base na pesquisa desenvolvida, várias vertentes para futuros trabalhos podem ser identificadas, como possíveis trabalhos futuros, pode-se apontar:

- Pesquisa para criar regras específicas para o modsecurity bem como outros sistemas de detecção de intrusão;
- Desenvolver um sistema de detecção de intrusão que possa fazer análise em modo gráfico e o bloqueio de ameças diretamente no *firewall*;

7. REFERÊNCIAS

ALECRIM, Emerson. **O usuário root**. Disponivel em: <https://www.infowester.com/linroot.php>. Acesso em: 10 de março de 2017.

ARANHA, Diego de Freitas. **Tomando o controle de programas vulneráveis a buffer overflow**. Disponivel em: <http://cic.unb.br/~rezende/trabs/buffer_overflow.htm>. Acesso em: 15 de março de 2017.

ASSUNÇÃO, Marcos Flávio. **Segredos do Hacker Ético**, 4.ed. Visual Books, 2012.

Carreiro, Júnior. **Arquivo de configuração do mod_security**. Disponivel em <https://www.vivaolinux.com.br/artigo/Arquivo-de-configuracao-do-mod-security>. Acesso em 29 de Julho de 2017.

CERT.br. **Estatísticas dos Incidentes Reportados ao CERT.br**. Disponivel em: <https://www.cert.br/stats/incidentes/>. Acesso em: 8 de março de 2017.

CEZAR, Luiz. **Instalando o ModSecurity no Debian + Apache2**. Disponivel em: <https://imasters.com.br/artigo/21948/redes-e-servidores/instalando-o-modsecurity-no-debian--apache2/?trace=1519021197&source=single>. Acesso em: 26 de março de 2017.

CHAVES, M. H. P. **Análise de Estado de Tráfego de Redes TCP/IP para Aplicação em Detecção de Intrusão**. Dissertação de Mestrado em Computação Aplicada – INPE, set 2002.

Cisco. **WHAT IS SNORT?**. Disponivel em: <https://s3.amazonaws.com/snort-org-site/production/document_files/files/000/000/116/original/Snort_rule_infographic.pdf?AWSAccessKeyId=AKIAIXACIED2SPMSC7GA&Expires=1489155537&Signature=7M3A%2BtfPhcCfa5u%2FxU4WV1dsOgI%3D/>. Acesso em: 10 de março de 2017.

DEBIAN. **FAIL2BAN**. Disponivel em: <https://servidordebian.org/pt/wheezy/security/brute_force_attack/fail2ban>. Acesso em: 10 de março de 2017.

Dicionario Informal. **Buffer**. Disponivel em: <http://www.dicionarioinformal.com.br/buffer/>. Acesso em: 15 de março de 2017.

DEVMEDIA. **Sistema de Detecção de Intrusão - Artigo Revista Infra Magazine 1**. Disponivel em: <http://www.devmedia.com.br/sistema-de-deteccao-de-intrusao-artigo-revista-infra-magazine-1/20819>. Acesso em: 13 de março de 2017.

FERREIRA, Ricardo. **Como proteger seu servidor Linux contra ataques automatizados usando a ferramenta Fail2Ban**. Disponivel em: <http://sysadmin.linuxdescomplicado.com.br/2016/05/como-proteger-seu-servidor-linux-contra-ataques-automatizados-usando-a-ferramenta-fail2ban/>. Acesso em: 10 de março de 2017.

Fidelis, Matheus. **ModSecurity Web Application Firewall: Configurando Regras de Bloqueio de SQL Injection e XSS**. Disponivel em <http://www.nanoshots.com.br/2016/01/modsecurity-web-application-firewall.html>. Aces so em 29 de Julho de 2017.

GIAVAROTO, Sílvio César Roxo; SANTOS, Gerson R. dos. **BacktTrack Linux e Teste de Invasão em Redes de Computadores**, 1. ed. Rio de Janeiro: Editora Ciência Moderna, 2013.

GUERRA, Lisandro. **O que é um servidor LAMP**. Disponivel em: <https://www.vivaolinux.com.br/artigo/Servidor-LAMP-no-Linux-Mint-e-Ubuntu/>. Acesso em: 20 de Julho de 2017.

HORA, Victor. **Home**. Disponivel em: <https://github.com/SpiderLabs/ModSecurity/wiki>. Acesso em: 8 de março de 2017.

HORA, Victor. **OWASP ModSecurity Core Rule Set (CRS) Project**. Disponivel em: <https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual#OWASP_ ModSecurity _Core_Rule_Set_CRS_Project>. Acesso em: 8 de março de 2017.

JAQUIER, Cyril. **FAIL2BAN**. Disponivel em: <https://www.fail2ban.org/wiki/index.php/Main_Page>. Acesso em: 8 de março de 2017.

KIM, S.S.; REDDY A. L. N. **A Study of Analyzing Network traffic as Images in Real-Time**. Department of Electrical Engineering – Texas A&M University, 2005.

KIZZA, J. M. **Guide to Computer Network Security. New York**, NY: Springer, 2005.

MAGNUS, Mischel. MODSECURITY 2.5. Packt Publishing, 2009.

MURINI, Cléber Taschetto. **Análise dos Sistemas de Detecção de Intrusão em Redes: Snort e Suricata Comparando com Dados da Darpa**. 2014. 58.Trabalho de Conclusão de Curso(mestrado) – Universidade Federal de Santa Maria, Rio Grande do Sul, Santa Maria, 2014.

Tanembaum, A. S. **Redes de Computadores**, 4ª edição. São Paulo: Saraiva, 2003.

OISF. Suricata. Disponivel em: <https://suricata-ids.org/>. Acesso em: 8 de março de 2017.

RISTIC, Ivan; FOLINI, Christian. **MODSECURITY HANDBOOK**, 2. ed. Feisty Duck, 2017.

SANTOS, Victor. **Sistemas de Detecção de Intrusões (IDS – Intrusion Detection Systems) usando unicamente softwares Open Source**. Disponível <https://seginfo.com.br/2010/06/21/sistemas-de-deteccao-de-intrusoes-ids-intrusion-detection-systems-usando-unicamente-softwares-open-source/#sistemas-de-deteccao-de-intrusao-baseados-em-host-hids>. Acesso em: 30 de Março de 2017.

Suricata. **What is Suricata**. Disponivel em: <http://suricata.readthedocs.io/en/latest/what-is-suricata.html>. Acesso em: 8 de março de 2017.

Symantec. **port scanning (verificação de porta)**. Disponivel em: <https://www.symantec.com/pt/br/security_response/glossary/define.jsp?letter=p&word=port -scanning>. Acesso em: 15 de março de 2017.

TrusWave. **ModSecurity**. Disponivel em: <https://modsecurity.org/crs/>. Acesso em: 14 de março de 2017.

TrusWave. **ModSecurity**. Disponivel em: <https://modsecurity.org/download.html>. Acesso em: 10 de março de 2017.