Detecção e Prevenção de Invasões em Redes Corporativas Utilizando o Package SNORT em Integração com o PFSENSE.

Guilherme Guedes Mantellis, Kaio Fellipe
Orientador: Diego Fiori de Carvalho

Curso de Sistemas de Informação – Centro Universitário UNIFAFIBE

Bebedouro –SP – Brasil

{sisunifafibe}@unifafibe.com.br

RESUMO: *Este artigo descreve a importância da segurança da informação em redes corporativas, citando algumas vulnerabilidades existentes em redes de computadores que possam ser utilizadas por crackers para causar a perda de informações estratégicas ou a indisponibilidade das mesmas por um período. Explicaremos alguns filtros de conteúdo utilizados por profissionais para diminuir os riscos destes ataques, comprovando sua ineficácia quando se trata de um ataque causado por uma vulnerabilidade de um serviço autentico liberado na rede. Para preenchimento desta vulnerabilidade, realizaremos a implementação do Snort no pfSense, um IDS/IPS que detecta e bloqueia o trafego malicioso na rede.*

Palavras-chave: Segurança da informação.*Firewalls*. IDS/IPS. *PfSense*. *Snort*.

1. INTRODUÇÃO

A informação se tornou o capital mais valioso de qualquer empresa, deixando-a sob constante risco. A intrusão de redes de computadores é um grande problema a ser enfrentado, estas intrusões ocorrem devido a vulnerabilidades em sistemas de informação, aplicativos, sistemas operacionais, dentre outros, onde *crackers* em sua maioria realizam tentativas para adentrar neste ambiente privado.

Várias soluções estão disponíveis no mercado para que possamos ter um melhor controle no trafego da rede, como a configuração de um *Firewall*, ou de um *proxy*, com isto é recomendado a utilização de um *Appliance*, que consegue integrar estas ferramentas em um lugar só, um exemplo disto é o *PfSense*, sendo uma ferramenta *OpenSource*, que possibilita uma administração completa de uma rede de computadores.

Contudo, mesmo com a obtenção destes filtros de conteúdo, ainda existe uma lacuna que *firewalls* ou *proxys* não conseguem preencher. Devido a sua limitação de filtragem, conseguindo somente bloquear ou liberar o trafego, atividades maliciosas vindas de *crackers* podem passar despercebidas por serviços confiáveis na rede, tornando cada vez mais necessário a utilização de softwares para detecção

e prevenção de invasão conhecidos como IDS e IPS, uma vez que estes consigam realizar esta filtragem com base no comportamento e conteúdo do tráfego, dando assim maior segurança no ambiente corporativo.

2 SEGURANÇA DA INFORMAÇÃO

A segurança da informação pode ser definida como a proteção existente sobre qualquer informação pessoal e corporativa. Podendo ser afetada por fatores comportamentais e de uso de quem se utiliza dela, pelo ambiente ou infraestrutura que a cerca e por pessoas mal intencionadas que tem o objetivo de obstrui-las, modificando, furtando e até mesmo destruindo tal informação. (SÊMOLA, 2003).

Como mencionado por Stallings (2014, p. 7) a definição de segurança da informação está voltada para "três objetivos principais que são o coração da segurança de computadores". Estes objetivos são normalmente chamados de tríade CIA (do acrônimo em inglês para *confidentiality*, *integrity e availability*). Estes conceitos são validos para a segurança tanto para os dados quanto para serviços de computação realizados. Ainda podem ser citados a autenticidade e o não repúdio da informação.

2.2 Vulnerabilidades de sistemas e redes de computadores

O site oficial do antivírus Norton (2016), explica que as vulnerabilidades podem ser definidas como falhas no *software* de computador que criam deficiências na segurança geral do computador e na rede, são fruto de um descuido ou *bug* de implementação que permite que alguém utilize esta falha para causar danos ao proprietário.

AOWASP Foundation (2017), realiza a atualização mensal de uma lista que consta as 10 vulnerabilidades mais vistas no ano. Dentre todas as vulnerabilidades conhecidas, esta lista traz itens que poderiam ser evitados adotando políticas de segurança simples e atualizações nos sistemas utilizados, além da adoção de ferramentas gratuitas para segurança em redes de computadores. O OWASP Top 10 conta com as seguintes vulnerabilidades demonstradas na Figura 1. Em uma atualização realizada no mês de março de 2017, uma vulnerabilidade foi adicionada à lista pela primeira vez indo direto para o *rank* A7, citada como *InsufficientAttackProtection* (Detecção e Prevenção de Ataques Insuficientes).

Fonte: OWASP *Foundation*, 2017 Top 10 *List*.

Figura 1. Top 10 Vulnerabilidades OWASP 2017.

2.2.1 *InsufficientAttackProtection*

Muitas aplicações e sistemas utilizados para segurança em uma rede de computadores não possuem a idoneidade básica para detectar, prevenir e refutar ataques manuais e automatizados. O resguardo contra ataques está longe de uma simples validação de entrada realizada por um *firewall*, ele envolve a detecção, o registro, e a resposta para estas tentativas de aproveitamento. (OWASP Foundation, 2017).

Considerando que qualquer pessoa com acesso à rede é capaz de enviar um ataque, aplicativos comuns de segurança conseguem detectar somente a entrada invalida da requisição, porém não realizando o bloqueio do endereço que as envia, deixando livre para realizar o uso de sondas em busca de vulnerabilidades, aumentando a probabilidade de exploração bem sucedida a 100%. Isso pode causar um impacto sobre o negócio, ataques bem sucedidos e não descobertos por longos períodos de tempo, expandem-se muito além do prejuízo inicial. (OWASP Foundation, 2017).

2.3 *Firewalls*

De acordo com Cheswisck, Bellovin e Rubin (2005, p. 205), “um *firewall* é qualquer dispositivo, software, arranjo ou equipamento que limita o acesso à rede.” Todo e qualquer dispositivo atualmente conta com um *firewall* gratuito integrado,

dentre estes dispositivos estão os populares roteadores e modems domésticos, utilizados por qualquer um que possua uma conexão com a *Web*.

Se um *firewall* for empregado indevidamente, “a única coisa que proporcionarão será um falso sentido de segurança”. *Firewalls* são comprovadamente ineficazes contra ataques internos de uma rede de computadores. (CHESWISCK, BELLOVIN E RUBIN, 2005, p. 206).

2.3.1 *pfSense*

O projeto *pfSense* é uma distribuição de *firewall* de rede livre, baseado na plataforma *FreeBSD* com um *Kernel* customizado, e inclui pacotes de *softwares* livres de terceiros para funcionalidades adicionais permitindo uma maior capacidade de expansão sem adicionar potenciais vulnerabilidades de segurança à distribuição base. O *software pfSense*®,com a ajuda do sistema de pacotes, é capaz de fornecer a mesma funcionalidade ou mais de *firewalls* comerciais comuns, sem nenhuma das limitações artificiais (PFSENSE RUBICON COMMUNICATIONS, 2017).

2.4 IDS/IPS

IDS é um sistema que monitora todos os pacotes que chegam a um site e notifica o administrador se uma violação de segurança é detectada, (COMER, 2009), porem o IDS não pode realizar nenhuma contra medida a qualquer tipo de ataque, agindo somente em modo passivo (*Promiscous-mode*), não interferindo no trafego da rede.

O IPS usa a capacidade de detecção do IDS junto com a capacidade de bloqueio de um firewall, notificando e bloqueando de forma eficaz qualquer tipo de ação suspeita ou indevida (ALENCAR, 2010).

2.4.1 *Snort*

O *Snort* é um sistema de detecção e prevenção de invasões de rede, agindo como um filtro, ele é capaz de realizar analises nos pacotes que estão trafegando em redes IP em tempo real. Ele realiza analises em protocolos de redes, busca de conteúdo automático para atualização de regras, além de ser usado para detectar uma variedade de ataques (SNORT AND CISCO, 2017).

2.5“*Jails”* em distribuições*Linux*

No Linux, podemos realizar a separação do usuário root dos usuários comuns, realizando a instalação e configuração de alguns programas no sistema, como exemplo possuímos o *LShell (Limited Shell)*, este irá realizar a limitação nos comandos que o usuário poderá utilizar no sistema, bloqueando o acesso dele a informações que somente o administrador pode ter acesso (FIDELIS, 2016).

3 METODOLOGIA

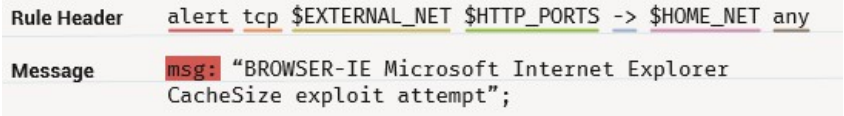
Realizaremos uma breve explicação das ferramentas utilizadas neste projeto, a metodologia tem início com o cerne do projeto, o *Snort*, uma poderosa ferramenta que realiza a análise de comportamento do tráfego da rede. Após isto, o ambiente em que esta ferramenta será aplicada, como sua topologia original, e topologia almejada após a implementação da *Jail* em conjunto.

3.1 Implementação do *Snort*

O pacote *Snort* opera por meio da utilização de assinaturas de detecção chamadas de *Rules* (Regras), podendo estas serem criadas pelos próprios usuários, ou simplesmente baixando um conjunto de regras pré-embaladas que o pacote *Snort* oferece suporte (HOW-TO SERIES PFSENSE, 2015).

3.1.1*Rules do Snort*

Para realizar a implementação do *Snort* em um ambiente como IPS, ou seja, interferindo no trafego da rede com a capacidade de bloqueio que ele possui, devemos entender como cada mensagem de alerta funciona, assim como as regras criadas para gerar estes alertas, para a identificação de falsos positivos e não deixando falsos negativos passarem despercebidos. Na Figura 2 temos a estrutura de uma regra do Snort.



Fonte: SNORTLOGY 101, The Anatomyof a SnortRule.

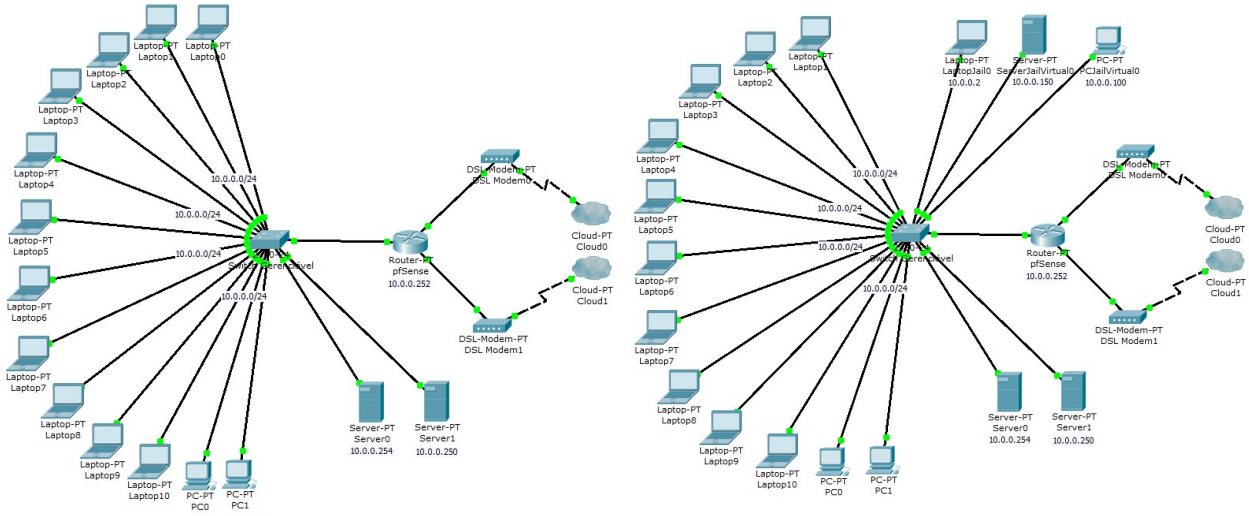
Figura 2. Características de uma Regra do *Snort*.

A primeira parte descrita como *Rule Header* contém os endereços IP e as máscaras de rede, protocolo, origem e destino da regra e as informações de portas

de origem e de destino. O primeiro item de uma regra é o alerta, ele informa ao *Snort*o que fazer quando encontra um pacote que se encaixe em um critério desta regra. O próximo campo em uma regra é o protocolo, existem 4 protocolos que o *Snort*consegue analisar comportamentos maliciosos, dentre estes estão o TCP, UDP, ICMP e IP. A parte descrita como $EXTERNAL_NET e $HTTP_PORTS são respectivamente nesta ordem, o endereço de IP de origem e a porta de origem. O operador de direcionamento (->) indica a orientação do trafego que a regra se aplica. E por último o $HOME_NET e *any*são respectivamente o endereço de destino e porta de destino. Logo em seguida é gerada uma simples mensagem para o usuário, ela geralmente contém o que a regra esta detectando. (SNORTOLOGY 101, 2015).

3.3 Ambiente de rede

Na Figura 3 é exibido à esquerda a topologia original desta rede corporativa, logo à direita a topologia desejada com a *Jail* presente nesta, o *pfSense* é encontrado entre os dois *modens* de *internet* e o *switch* gerenciável principal. A rede original conta com 16 *hosts* físicos, sendo dois servidores físicos (*Server0*[IP 10.0.0.254] e *Server1*[IP 10.0.0.250]), o servidor onde o *pfSense* está operando (*Router*-PT *pfSense*[IP 10.0.0.252]), e os demais *hosts* são destinados aos diferentes setores da empresa.



Fonte: Elaborada pelos autores.

Figura 3. Topologia de rede Original x Almejada.

A rede almejada à direta possui a mesma quantidade de *hosts* físicos que a rede original. O diferencial está nos dois *hosts* virtualizados acima hospedados no

LaptopJail0 (já existente na rede original), simulando uma rede falsa mesmo sendo roteada pelo mesmo *gateway* (*pfSense*). Explicaremos mais a respeito na sessão abaixo.

4 DESENVOLVIMENTO

O *Appliance pfSense* vem sendo utilizado por empresas que necessitam e desejam possuir um controle sobre as informações que transitam pela sua rede de computadores.

Com a grande limitação dos *firewalls* convencionais utilizados, não possuindo a filtragem de conteúdo malicioso por serviços confiáveis na rede, softwares como o *Snort* vem surgindo com grande força em ambientes corporativos, uma vez que consigam preencher esta lacuna deixada pelos *firewalls*, *proxies*, etc.

4.1 Estratégia

Ao refletirmos sobre como iria ser feita a implementação da *Jail* neste ambiente, nos deparamos com diversos paradigmas. Necessitaríamos da isolação da rede LAN, para que a mesma não sofresse influência da rede *Jail*, realizamos um levantamento nas diferentes hipóteses que nos cercavam.

4.1.1 *Switch* Gerenciável com VLAN

Neste cenário, utilizaríamos do Switch Gerenciável presente na rede, para realizar a configuração de uma VLAN, onde o *pfSense* seria o gerenciador destas VLANs, separando assim a rede LAN principal, da rede *Jail*, realizando o isolamento da rede, conseguindo controlar o seu tráfego no *pfSense*.

Ao utilizarmos estas configurações, há a possibilidade de descoberta da VLAN pelo invasor, caso o mesmo analise o tráfego pelo *wireshark* e veja que o pacote está marcado para uma VLAN.

4.1.2 *Switch* Gerenciável com *portforward*+ *LShell*

O melhor caminho traçado para realizarmos estas configurações colocando-as em produção de maneira ágil, foi a utilização do pacote *LShell* no *Linux*, onde seria realizado uma limitação nos comandos que poderiam ser utilizados no terminal do *Linux*, “enganando” por assim dizer, quem utilizasse este usuário. Alinhando isto com as configurações de *portforward* no *Switch*, separando este *host* de qualquer conexão com a rede LAN, possuiríamos em mãos, a *Jail*.

4.2 Configurações

No processo de criação deste ambiente, nos deparamos com inúmeras configurações possíveis em cada um dos equipamentos/*softwares* utilizados, demonstraremos nesta sessão as configurações utilizadas, não especificaremos configurações adicionais devido as inúmeras possibilidades disponíveis, assim como campos que compõe informações sigilosas da empresa.

4.2.1*pfSense*

O *pfSense* manteve suas configurações padrões de fábrica, foram adicionados somente novas regras de *firewall* com *aliases* de endereços IPs e de domínio respectivamente em: *Firewall*> Regras; *Firewall*>*Aliases*.

Para realizar a conexão SSH com o servidor de testes utilizado realizamos a criação de uma regra de NAT da porta 22(padrão SSH) para o endereço IP deste servidor em: *Firewall*> NAT > Encaminhamento de Portas. A regra criada contém as seguintes especificações:

- Interface: Interface de Entrada do tráfego (SINALBR no nosso caso)
- Protocolo: TCP
- Endereço de Origem: * (qualquer)
- Portas de Origem: * (qualquer)
- Dest. Address: * (qualquer)
- Dest. Ports: 22 (SSH)
- IP NAT: IP do host que será atacado
- Portas NAT: 22 (SSH)
- Descrição: Descrição desejada para a regra

Após isto foi adicionado o *packageSnort* diretamente da lista de pacotes oficiais do *pfSense* em: Sistema > Gerenciador de Pacotes > Pacotes disponíveis.

4.2.2*Snort*

Buscando agilidade na instalação inicial, o *packageSnort*dá a opção de seleção de assinatura de regras por categoria, possibilitando a implementação do mesmo sem o conhecimento prévio das regras. No caminho: Serviços >*Snort*>*Categorias*> Interface desejada.

As possibilidades de seleção de categoria de regras são 3: *Connectivity*(Conectividade), *Balanced*(Balanceamento) e *Security* (Segurança). É recomendado utilizar a categoria para conectividade no início da elaboração, para uma adaptação as regras existentes no *Snort*. (HOW-TO SERIES PFSENSE, 2015).

Após este período inicial de adaptação as regras que o *Snort*possui, deve-se começar a análise dos alertas gerados no caminho:Serviços >*Snort*>*Alerts,*para identificação dos falsos positivos presentes antes da habilitação da função de bloqueio do *Snort.* Para isto a identificação de serviços vitais para o acesso da empresa deve ser feita, adicionando-os a uma *PassList* (Lista de passes). (HOW-TO SERIES PFSENSE, 2015).

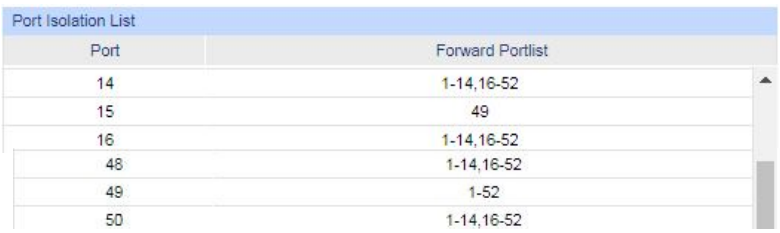
Para habilitar a função de bloqueio do *Snort*, devemos habilitar esta opção no caminho: Serviços >*Snort*> Configuração Geral > marcar a opção para realizar o bloqueio. Com isto, o *Snort* além de realizar a detecção de tráfego malicioso na rede, também irá bloquear este tráfego.

4.2.3 Switch Gerenciável

Com as possibilidades de configurações do *switch* gerenciável, a administração de redes de computadores se tornou algo pratico, onde conseguimos informações de todas as portas físicas do *switch*, quais estão ativas, qual computador está conectado a esta porta, as delimitações de velocidade de rede de cada conexão, entre outras possibilidades.

4.2.3.1 *PortForward*

Esta configuração nos permite isolar hosts conectados a portas especificas, permitindo ou não a sua comunicação com outras portas do *Switch*. Como podemos ver na Figura 4, realizamos o isolamento da porta 15 que possui o host *Jail*, conseguindo enviar e receber pacotes somente do host conectado à porta 49(*gatewaypfSense*). As demais portas (1 – 13 e 17 – 47) possuem as mesmas configurações das portas: 14, 16, 48 e 50.


Port Isolation List

Port	Forward Portlist
14	1-14,16-52
15	49
16	1-14,16-52
48	1-14,16-52
49	1-52
50	1-14,16-52

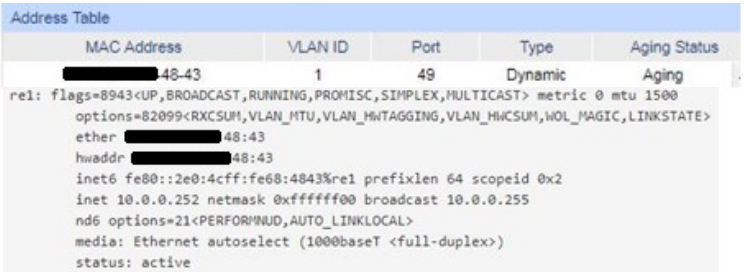
Figura 4. Configurações de *portforward*. Fonte: Elaborada pelos autores.

4.2.3.1 Comprovando configurações

Abaixo demonstraremos nas Figuras 5 e 6 o MAC *address* dos *hostsJail*(máquina física) e *pfSense* vinculados às suas respectivas portas, juntamente com suas configurações de rede local. Na Figura 5 temos o *host* físico da *Jail* que possui o MAC *address* de final 2C-43 conectado à porta 15 com o IP 10.0.0.2, utilizando o *gateway* 10.0.0.252. Na Figura 6 temos o *pfSense* com o MAC *address* de final 48-43 conectado à porta 49 com o IP 10.0.0.252, sendo o *gateway* da rede LAN.

Fonte: Elaborada pelos autores.

Figura 5. Configurações de porta e detalhes de conexão local *host* físico *Jail*.



Fonte: Elaborada pelos autores.

Figura 6. Configurações de porta e detalhes de conexão local do *pfSense*.

4.2.4 Servidor Linux

Utilizamos um servidor *Ubuntu* para realizar os testes de acesso externo. Realizamos a criação de dois usuários de teste, com possibilidades de acesso administrativo ao sistema (podendo executar comandos com o prefixo *sudo*).

4.2.4.1 SSH

Necessitamos somente de 2 passos para realizar a configuração do acesso SSH ao servidor.

1. Instalar o *OpenSSH* com o seguinte comando: #sudo apt-get install openssh-server;
2. Bastando agora parar o serviço SSH e iniciar ele novamente, respectivamente com estes comandos: #sudo/etc/init.d/ssh stop e #sudo/etc/init.d/ssh start.

4.2.4.2 *LShell*

Após realizar a criação dos usuários de teste, podemos iniciar com a instalação e configuração do *LShell*, seguimos esta sequência para realizar isto:

1. Realizamos a instalação do *LShell* com o comando: #sudo aptitude install shell;
2. Para alterar as configurações de acesso, como comandos permitidos dos usuários, local onde será gravado os logs, etc., precisamos abrir o arquivo lshell.conf com o seguinte comando: #sudogedit /etc/lshell.conf;
   As linhas alteradas no arquivo foram as seguintes:

```
# lshell.py configuration file
#
# $Id: lshell.conf,v 1.27 2010-10-18 19:05:17 ghantoosExp $
[global]
allowed: ['ls']
forbidden: ['apt-get', 'install']
sudo_commands: ['ls']
intro: "UbuntuServer02\n"
```

2.1. As configurações alteradas no arquivo são as seguintes:

No campo *allowed*, constam os comandos permitidos para o usuário executar. No campo *forbidden foram adicionados comandos* proibidos para execução. Na linha *sudo_commands*, consta os comandos que podem ser executados como administrador.

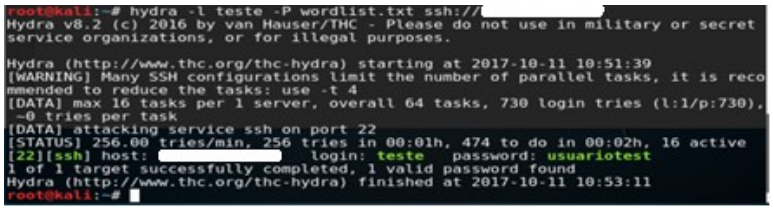
3. Restando somente adicionar o usuário que irá utilizar o *LShell,* para isto basta executar o seguinte comando: #sudo usermod -s /usr/bin/lshellusuario.

5 RESULTADOS

Nesta sessão serão apresentados os resultados obtidos com os testes realizados, demonstraremos o *Snort* bloqueando um teste de força bruta no serviço SSH, comprovando sua capacidade, assim como a rede *Jail* demonstrando sua eficácia isolando o atacante em um espaço restrito, fora da rede convencional.

5.1 Detecção de *BruteForce* SSH (IDS)

Para a realização deste teste, utilizamos a ferramenta *Hydra* disponível no *KaliLinux*, como mostrado na Figura 7 o ataque de força bruta foi bem sucedido com a utilização de uma *wordlist* contendo 730 senhas, encontrando a senha de acesso ao usuário teste.



Fonte: Elaborada pelos autores.

Figura 7. Ataque de força bruta com a ferramenta *Hydra* (*Snort* habilitado para detectar o ataque somente).

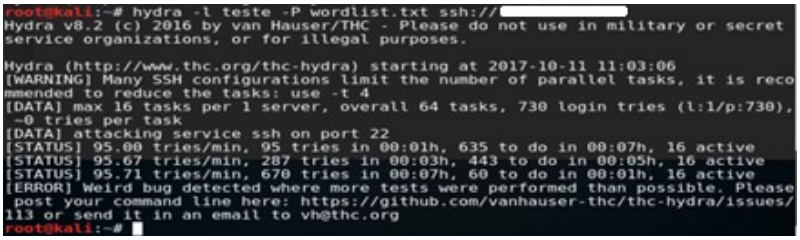
No mesmo contexto do ataque mencionado acima, abaixo na Figura 8 possuímos os alertas gerados pelo *Snort* no momento do ataque, podemos notar que o ataque teve início no dia 11/10/2017 às 10:51:39, o *Snort* detectou este ataque 3 segundos após o seu início devido a distância entre os dois *hosts,* ou seja, o tempo de detecção do *Snort* é instantâneo, a partir do momento que o tráfego chega a ele.

Fonte: Elaborada pelos autores.

Figura 8. Alertas gerados pelo *Snort* no momento do ataque de força bruta.

5.2 Prevenção de *BruteForce* SSH (IDPS)

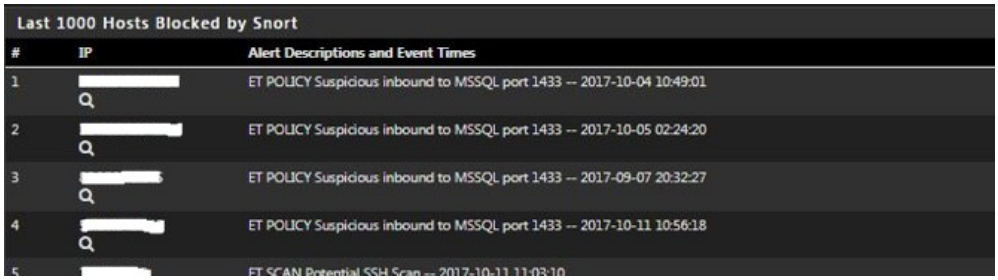
Neste ataque utilizamos as mesmas configurações do anterior na seção 5.1, com uma única diferença, o *Snort* estava habilitado para realizar o bloqueio caso algum alerta fosse gerado. Na Figura 9 podemos verificar que não foi encontrado a senha de acesso do usuário teste como no exemplo acima, a *Hydra* realizou o teste com todas as 730 senhas na *wordlist*, porem mesmo assim não foi encontrada.



Fonte: Elaborada pelos autores.

Figura 9. Ataque de força bruta com a ferramenta *Hydra* (*Snort* habilitado para bloquear o ataque).

Na Figura 10 podemos notar que o host que estava realizando o ataque ao servidor SSH está na lista de bloqueio do *Snort*, sendo assim, qualquer conexão que este host possuía com a rede que o servidor SSH está, foi cortada 4 segundos após o início do ataque, sendo esta a causa da falha ao encontrar a senha de acesso para o usuário teste.



Fonte: Elaborada pelos autores.

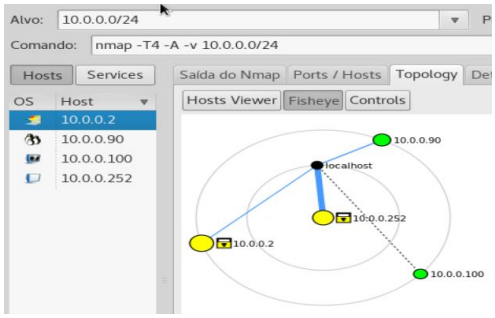
Figura 10. Tabela de bloqueios do *Snort*.

5.2*Jail*

A segunda linha de defesa utilizada para dar a ilusão ao atacante de ter entrado em um servidor real é a rede *Jail* como explicada nos tópicos acima. Realizamos testes de verificação com o *Nmap* buscando os serviços e hosts disponíveis na rede.

5.2.1 Verificação dos *hosts* com *Nmap*

Com a utilização do *Nmap* conseguimos verificar os hosts que estão operantes na rede. Como podemos visualizar na Figura 11, a topologia apresentada da rede, mostra somente os hosts 10.0.0.2 (hospedeiro do servidor SSH[10.0.0.90] e do *KaliLinux*[10.0.0.100]) e o 10.0.0.252(*gateway* da rede).



Fonte: Elaborada pelos autores.

Figura 11. Topologia de rede encontrada pelo *Nmap*.

5.2.2 Limitação de comandos

Conseguimos um controle alto das ações que o invasor poderia realizar utilizando o *LShell*, como mostrado no quadro 1, a esquerda podemos ver os comandos que o invasor tentou utilizar, comandos que ele não possuía acesso, foram bloqueados e obtiveram o seguinte retorno: “#***forbiddensyntax: (comando utilizado)”.

Diferente da esquerda do quadro, à direta podemos notar que os retornos dos comandos utilizados para um usuário comum, também com privilégios de administrador, se diferenciam bastante. Não possuindo nenhum bloqueio.

QUADRO 1 – Comparação entre usuário teste à esquerda (utilizando a limitação do *LShell*) e o usuário teste2 à direita (não utilizando a limitação do *LShell*).

Login* com Usuário utilizando *LShell	***Login* com Usuário sem o *LShell***

#login as: teste #teste@---------- password: #Lastlogin :WedOct 11 16:07:25 2017 from XXX.XXX.XXX.XXX #UbuntuServer02 **#teste:~$ ls** #Área de Trabalho Downloads Imagens Música Vídeos Documentos examples.desktop Modelos Público **#teste:~$ sudosu** #***forbiddensyntax: sudosu **#teste:~$ sudols** #[sudo] senha para teste: #Área de Trabalho Downloads Imagens Música Vídeos Documentos examples.desktop Modelos Público	**#login as: teste2** #teste2@---------- password: #Lastlogin :WedOct 11 16:12:26 2017 from XXX.XXX.XXX.XXX **#teste2@server01-VirtualBox:~$ ls** #Área de Trabalho Downloads Imagens Música Vídeos Documentos examples.desktop Modelos Público **#teste2@server01-VirtualBox:~$ sudosu** #[sudo] senha para teste2: **#root@server01-VirtualBox:/home/teste2# apt-getupdate** #Obter:4 http://security.ubuntu.com/ubuntu zesty-securityInRelease [89,2 kB]

Fonte: Elaborado pelos autores.

6 CONSIDERAÇÕES FINAIS

Com os testes realizados conseguimos notar o grande potencial do *Snort* ao realizar a detecção e bloqueio de ataques passivos e ativos, com suas regras sendo atualizadas semanalmente, a maioria das vulnerabilidades presentes em redes de computadores são protegidas por este sistema, impedindo o aproveitamento das mesmas por usuários maliciosos dentro e fora de uma rede corporativa.

A utilização da rede *Jail* em conjunto com o *Snort* nos possibilitou uma maior segurança da informação no ambiente corporativo, conseguindo analisar o comportamento de um invasor de forma segura. Como trabalhos futuros, sugerimos a expansão do *Snort* para redes corporativas de maior porte.

REFERÊNCIAS

CHEWISCK, William R.; BELLOVIN, Steven M.; RUBIN, Aviel D. **Firewalls e Segurançana Internet**. 2ª. ed. Porto Alegre: Bookman, 2005. 400 p.

STALLINGS, William. **Criptografia e segurança de redes**:Princípios e Práticas. 6ª. ed. São Paulo: Pearson Education do Brasil, 2014. 558 p.

COMER, Douglas E. **Redes de computadores e internet**. 4ª. ed. São Paulo: Bookman, 2007. 632 p.

SÊMOLA, M. **Gestão da Segurança da Informação**. Rio de Janeiro: Campus, 2003.

ALENCAR, Leandro. Diferença entre IDS e IPS. Disponívelem: <http://forsecurity.blogspot.com.br/2010/06/diferenca-entre-ids-e-ips.html>. Acessoem: 05 abr. 2017.

FIDELIS, Matheus. Criando uma shell limitada para o usuário no Linux. Disponível em: <http://www.nanoshots.com.br/2016/09/lshell-criando-uma-shell-limitada-para.html>. Acesso em: 14 de outubro de 2017.

How-ToSeries pfSense. Setup SnortPackage. Disponível em <https://doc.pfsense.org/index.php/Setup_Snort_Package>. Acesso em: 14 de outubro de 2017.

Kali Linux Official Documentation. What is Kali Linux?.Disponívelem: <http://docs.kali.org/introduction/what-is-kali-linux>. Acessoem: 20 maio 2017.

OWASP Foundation. OWASP Top Ten 2017 Project. Disponívelem: <https://www.owasp.org/index.php/Top_10_2017>. Acessoem: 03 maio 2017.

Rohr by Symantec. Como elesatacam: Vulnerabilidades. Disponívelem: <https://br.norton.com/security_response/vulnerabilities.jsp>. Acessoem: 25 maio 2017.

SNORTOLOGY 101, The Anatomy of a Snort Rule. Disponívelem: <https://snort.org/documents#OfficialDocumentation>. Acessoem: 25 maio 2017.